

:login:

The USENIX Association Newsletter

Volume 14, Number 6

November/December 1989

CONTENTS

1990 Elections for Board of Directors	3
USENIX Winter Conference Program	4
Call for Papers: Summer 1990 USENIX Conference	10
Executive Office Staff Changes	11
Call for Papers: USENIX C++ Conference	12
EUUG Conferences	13
Call for Papers: AUUG Conference and Exhibition 1990	14
Long-Term Calendar of UNIX Events	15
USENIX Board Studies UUCP	16
Summary of Board of Directors' Meeting	
Audio I/O with the NeXT Computer	19
<i>Michael Hawley</i>	
Book Review: !%@:: A Directory of Electronic Mail Addressing and Networks	24
<i>Peter H. Salus</i>	
Report on ISO/IEC JTC1/SC22/WG15 (POSIX) Meeting	25
<i>Dominic Dunlop</i>	
An Update on UNIX and C Standards Activity	31
<i>Jeffrey S. Haemer</i>	
Proceedings Order Forms	46
Final Printing of 4.3BSD Manuals	48
4.3BSD Manual Reproduction Authorization and Order Form	49
Local User Groups	50

The closing date for submissions for the next issue of *:login:* is January 3, 1990.



THE PROFESSIONAL AND TECHNICAL
UNIX® ASSOCIATION

NOTICE

:login: (ISSN 1044-6397) is published bimonthly by the USENIX Association.

The USENIX Association is a not-for-profit organization of those interested in UNIX and UNIX-like systems. It is dedicated to fostering and communicating the development of research and technological information and ideas pertaining to advanced computing systems, to the monitoring and encouragement of continuing innovation in advanced computing environments, and to the provision of a forum where technical issues are aired and critical thought exercised so that its members can remain current and vital.

The officers of the Association are:

President	Alan G. Nemeth <i>agn@usenix.org</i>
Vice-President	Deborah K. Scherrer <i>scherrer@usenix.org</i>
Secretary	Rob Kolstad <i>kolstad@usenix.org</i>
Treasurer	Stephen C. Johnson <i>scj@usenix.org</i>
Directors	M. Kirk McKusick <i>mckusick@usenix.org</i> Sharon Murrel <i>eowyn@usenix.org</i> Michael D. O'Dell <i>mo@usenix.org</i> John S. Quarterman <i>jsq@usenix.org</i>
Executive Director	Ellie Young <i>ellie@usenix.org</i>

USENIX Association Office
2560 Ninth Street, Suite 215
Berkeley, CA 94710
(415) 528-8649
office@usenix.org

The editorial staff of *:login:* is:

Editor	Ellie Young
Managing Editor	Carolyn Carr
Copy Editor	Michelle Dominijanni
Production Editor	Tom Strong

[†]UNIX is a registered trademark of AT&T.

[‡]SUN is a trademark of Sun Microsystems, Inc.

^{*}Lasergrafix is a trademark of Quality Micro Systems.

[#]Sybase is a registered trademark of Sybase, Inc.

Judith F. DesHarnais Conference Coordinator

USENIX Conference Office
22672 Lambert Street, Suite 613
El Toro, CA 92630
(714) 588-8649
judy@usenix.org

John L. Donnelly Tutorial & Exhibit Manager

USENIX Exhibit Office
5398 Manhattan Circle
Boulder, CO 80303
(303) 499-2600
johnd@usenix.org

Contributions Solicited

Members of the UNIX community are encouraged to contribute articles to *:login:*. Contributions may be sent electronically to *login@usenix.org* or through the U.S. mail to the Association office. The USENIX Association reserves the right to edit submitted material.

:login: is produced on UNIX systems using *troff* and a variation of the -me macros. Contributions should be in *n/troff* input format, using any macro package.

UUNET Subscriptions

UUNET Communications
3110 Fairview Park Drive, Suite 570
Falls Church, VA 22042
(703) 876-5050
uunet-request@uunet.uu.net

Acknowledgments

The Association uses a SUN[‡] 3/180S running SUN OS for support of office and membership functions, preparation of *:login:*, and other Association activities. Connected to the SUN is a QMS Lasergrafix^{*} 800 Printer System donated by Quality Micro Systems of Mobile, Alabama. It is used for general printing and draft production of *:login:* and *Computing Systems* with *ditroff* software provided by mt Xinu. The membership and mailing lists are maintained using Sybase.[#]

This newsletter is for the use of the members of the USENIX Association. Any reproduction of this newsletter in its entirety or in part requires written permission of the Association and the author(s).

1990 Elections for Board of Directors

The biennial elections of the Association will be held in the Spring of 1990.

After the D.C. Conference, nominations from the membership will remain open until February 2, 1990. The procedure for nominations by the membership is a written statement of nomination signed by at least five (5) members in good standing (or five separate nominations), to be submitted to the Executive Director at the Association office, and received by noon, PST, February 2. Please include a Candidate's Statement for inclusion with the ballots as well.

Ballots will be sent to all paid-up members as of March 1, 1990, on or about March 12.

Report of the Nominating Committee

A nominating committee was chartered by the current Board of Directors in accordance with the By-Laws of the Association to nominate a slate of candidates for the upcoming election of Directors and Officers. The committee's charge was to ensure that there were at least as many suitable candidates nominated as there are positions on the Board. The committee solicited suggestions for nominees, interviewed all of those suggested plus several other people, and have nominated the people listed below. All of these nominees

Members will have until April 6 to return their ballots, in the envelopes provided, to the Association office. The results of the election will be announced at the Anaheim Conference and in the May/June issue of *:login:*.

The Board is comprised of eight directors, four of whom are "at large." The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential, with those candidates with the largest number of votes being elected. Newly elected directors will take office immediately following the Anaheim conference in June.

want to serve on the Board and have indicated to the committee that they have the support of both their employers and families for the time commitment involved.

There are certainly many other qualified candidates. The committee did not attempt to nominate all of the potentially good Board members, but nominated what it felt to be a good slate of candidates. Any member of the Association may be nominated by petition for any board position (see above instructions).

The candidates nominated by the committee are:

President	Steven C. Johnson, Stardent Computer
President	Marshall Kirk McKusick, University of California
Vice President	Michael D. O'Dell, Prisma, Inc.
Secretary	Rob Kolstad, Prisma, Inc.
Treasurer	Sharon Murrel, AT&T Bell Laboratories
Director	Peter Collinson, Hillside Systems
Director	Ed Gould, mt Xinu
Director	Daniel Klein, Software Engineering Institute, Carnegie Mellon University
Director	Evi Nemeth, University of Colorado
Director	Sonya D. Neufer, Canstar
Director	Barry Shein, Software Tool and Die
Director	Dave Taylor, Intuitive Systems
Director	Alix Vasilatos, Open Software Foundation

Members of the nominating committee are:

Ed Gould, mt Xinu, Chair
Tom Ferrin, Univ. of California, San Francisco
Charlie Sauer, Dell Computer

Wendy Thrash, University of Washington
Pat Wilson, Consultant
Elizabeth Zwicky, SRI International

USENIX Winter Conference Program

Omni Shoreham Hotel, Washington, D.C., January 22-26, 1990

Tutorials

Monday, January 22

UNIX on Modern Architectures

Curt F. Schimmel, Amdahl, Key Computer Labs

Creating User Interfaces with OSF/Motif

Kee Hinckley & Brian Holt, Apollo Computer, Inc.

UNIX Network Programming

Richard Stevens, Health Systems International

Introduction to 4.3BSD Internals

Thomas W. Doeppner, Jr., Brown University

UNIX System V Release 4.0 Internals - Introduction

Steve Buroff & Mike Scheer, AT&T

Mach Overview

Avadis Tevanian, Jr., NeXT, Inc.

An Introduction To C++

Robert Murray, AT&T Bell Laboratories

Introduction To Programming the X Window System,* Version 11

Oliver Jones, HP Apollo Systems Division

Tuesday, January 23

An Introduction to Object-Oriented Programming

David Taenzer, U.S. West Advanced Technologies

Open Systems Interconnection (OSI) Principles

Colin I'Anson, Hewlett Packard Laboratories

Software Contracts and Intellectual Property

Daniel Appelman, Heller, Ehrman, White & McAuliffe

Beyond 4.3BSD: Advanced Kernel Topics

Mike Karels & Marshall Kirk McKusick, University of California, Berkeley

Topics in System Administration

Rob Kolstad, Prisma Inc., & *Evi Nemeth*, University of Colorado

Mach Virtual Memory Internals

Nawaf Bitar, Hewlett-Packard Company

Using C++ Effectively

Andrew Koenig, AT&T Bell Laboratories

X Toolkit Intrinsic

Paul E. Kimball, Digital Equipment Corporation

Special Note for Full Time Students: A limited number of spaces in each tutorial class have been reserved for full time students at a special fee. Please contact the Conference office for full details.

* The X Window System is a trademark of M.I.T.

Technical Conference Program

Wednesday, January 24

9:00-10:30 Introductory Remarks

*Daniel Klein, Software Engineering Institute, CMU
Ellie Young, USENIX Association*

KEYNOTE: NASA's Manned Spacecraft Computers

Jim Tomayko, Software Engineering Institute, CMU

10:30-11:00 Break

11:00-12:30 Virtual Memory

Chair: Chet Juszczak

A Dynamic File System Inode Allocation and Reclaim Policy

Ron Barkley & T. Paul Lee, AT&T Bell Laboratories

Insuring Improved VM Performance: Some No-Fault Policies

Danny Chen, Ron Barkley, & T. Paul Lee, AT&T Bell Laboratories

**An External Pager Implemented as a UNIX System V,
Release 4 Virtual File System**

Dean Thomas, Unisys Corporation

12:30- 2:00 Lunch

2:00- 3:30 Architecture & Debuggers

Chair: John Mashey

Implementing a Mach Debugger for Multithreaded Applications

*Deborah L. Caswell, Hewlett Packard Company,
David L. Black, Carnegie Mellon University*

pdb: A Network Oriented Symbolic Debugger

Paul Maybee, Solbourne Computer, Inc.

Some Efficient Architecture Simulation Techniques

Robert Bedichek, University of Washington

3:30- 4:00 Break

4:00- 5:30 Applications

Chair: Susanne Smith

Software Tickerplants on UNIX

*Mark Luppi, Robert Berkley, Skip Gilbrech,
Tim Hunt, & Richard Plevin, Fusion Systems Group*

GENESIS and XODUS – General Purpose Neural Network Simulation Tools

*John Uhley, U. S. Bhalla, M. A. Wilson, D. H. Bilitch,
M. E. Nelson, & J. M. Bower, California Institute of Technology*

Keynote – A Language and Extensible Graphical Editor for Music

Tim Thompson, AT&T Bell Laboratories

Thursday, January 25

9:00-10:30 **Utilities** Chair: John Devitofranceschi

Integrated Interactive Access to Heterogeneous Distributed Services

Joel S. Emer & William E. Weihl

MIT Laboratory for Computer Science

The UNIX System Math Library, A Status Report

Joel Silverstein, Steve Sommars, & Yio-Chian Tao

AT&T Bell Laboratories

Tcl: An Embeddable Command Language

John K. Ousterhout, University of California, Berkeley

10:30-11:00 Break

11:00-12:30 **Kernel Internals**

Chair: Charlie Perkins

An Event-based Fair Share Scheduler

Raymond B. Essick, Prisma, Inc.

Parallel STREAMS: a Multi-Processor Implementation

Arun Garg, Sequent Computer Systems

Implementing Berkeley Sockets in System V, Release 4

Ian Vessey & Glenn Skinner, Sun Microsystems

12:30- 2:00 Lunch

2:00- 3:30 **Networks**

Chair: Alix Vasilatos

Two Network Management Tools -or- (How Many Packets Would a
Packet Router Route if a Packet Router Could Route Packets?)

Jeff Okamoto & Allan Lienwand, Hewlett Packard Company

Packet Trains on NSFNET National Backbone – A Traffic Characterization

Steven A. Heimlich, University of Maryland

Pseudo-Network Drivers and Virtual Networks

Steven Bellovin, AT&T Bell Laboratories

3:30- 4:00 Break

4:00- 5:30 **Ethics in the Computer Industry**

Moderator: Rob Kolstad

A panel composed of a lawyer, a CEO, an ethicist and others will discuss various questions about ethics in the computer industry.

Friday, January 26

9:00-10:30	User Interface Management Systems	Chair: Dan Geer
	The Serpent User Interface Management System <i>Brian Clapper, Erik Hardy, Rick Kazman, & Robert Seacord,</i> Software Engineering Institute	
	Parallel Object-Oriented UIMS with Macro and Micro Stubs <i>Masami Hagiya & Kouji Ohtani, Kyoto University</i>	
	MTX – A Shell that Permits Dynamic Rearrangement of Process Connections and Windows <i>Stephen A. Uhler, Bell Communications Research</i>	
10:30-11:00	Break	
11:00-12:30	File Systems	Chair: Kirk McKusick
	Using UNIX as One Component of a Lightweight Distributed Kernel for Multiprocessor File Servers <i>David Hitz, Guy Harris, James Lau, & Allan Schwartz,</i> Auspex Systems Inc.	
	A Highly-Parallelized Mach-based Vnode Filesystem <i>Alan Langerman, Joseph Boykin, Susan LoVerso, & Shashi Mangalat,</i> Encore Computer Corporation	
	Disk Scheduling Revisited <i>Margo Seltzer, Peter Chen, & John Ousterhout,</i> University of California, Berkeley	
12:30- 2:00	Lunch	
2:00- 4:00	Languages & Software Engineering	Chair: Dan Klein
	Postloading for Fun and Profit <i>Stephen C. Johnson, Ardent Computer Corporation</i>	
	Multiple Site Source Reconciliation <i>Dodi Francisco & Lois C. Price, TRW Financial Systems, Inc.</i>	
	CVS-II: Parallelizing Software Development <i>Brian Berliner, Prisma, Inc.</i>	
	Ada and Binary UNIX Standards <i>Mitchell Gart, Alsys Inc.</i>	

New Concurrent Sessions

USENIX is pleased to introduce a new component to its technical conference. These experimental concurrent sessions will enable people to exchange ideas and information in a more informal atmosphere. Attendees will be free to migrate between all sessions. If there is sufficient interest, these new sessions will continue as a regular event.

Wednesday, January 24

11:00-12:30 Regular Expressions

Andrew Hume, AT&T Bell Laboratories

The general history of regular expressions, the best known algorithms at this time, and the history of regular expressions on UNIX will be discussed. The different types of regular expression syntaxes used by various UNIX commands (*sh, ed, lex, grep* etc.) will be examined and examples given of their use.

make

Andrew Hume

This talk is a tutorial for generic *make*, including macros and built-in rules. Also included are some dirty tricks and discussion of various other makes.

2:00- 3:30 Submitting and Presenting Papers at USENIX

This talk will give you clues on getting your paper accepted: what we look for and why we accept or reject papers, as well as offering suggestions on alternative places to submit papers. It will also cover what happens once your submission has been accepted: how you can ensure that your paper looks good in the proceedings, and hints for giving a good talk at the conference. This talk is given by a group of people who have been active in USENIX for several years.

Thursday, January 25

11:00-12:30 Getting the Most from Support

Mary Seabrook, UniSoft Corporation

Buying a support contract isn't enough. As a technical person, you need to learn how to use support as effectively as possible. This session describes how best to present your problem to enable your support department to find a solution. This includes some thoughts on how to detail the problem and information that may be most useful in tracking down bugs.

Surviving in Networkland

John Quarterman, Texas Internet Consulting

This is a brief overview of some of the principal networks you can reach by electronic mail from an average UNIX machine, some hints on how to do that, and some of the uses that you might want to make.

2:00- 3:30 ***nawk - A New Version of awk***

Richard Stevens, Health Systems International

This talk describes the differences between *awk* and *nawk*, patterns and regular expressions, flow control, expressions, variables and functions, input/output capability, and interaction with shells.

4:00- 5:30 **Works-in-Progress Session**

Chair: Clement Cole

Ten minute presentations of current work.

Friday, January 26

11:00-12:30 **Perl - A System Administration Language**

Tom Christiansen, Convex Computer Corporation

Perl is an interpreted language specifically designed for system administrators. In this talk it will be introduced and an overview of the syntax, as well as some examples of its use, will be given.

2:00- 4:00 **Works-in-Progress Session**

Chair: Michelle Dominijanni

Ten minute presentations of current work.

Terminal Room at D.C. Conference

The Winter USENIX Conference will once again have a terminal room providing Internet and dialout access for attendees to touch base and read their mail. Attendees will have to pay their own long distance charges by using an AT&T, MCI, Sprint, or another phone credit card. Local calls, however, will be free!

Facilities will be available to create cartridge tapes of miscellaneous, GNU, and public domain software.

During the conference electronic mail sent to *John_Doe@conference.usenix.org* will be printed on a laser printer in the terminal room and posted on the USENIX Message Board.

Many thanks to our terminal room sponsors: AT&T, Encore/Xylogics, IBM, OSF, QMS, Sun, and Telebit.

Sonya Neufer
USENIX Terminal Room Coordinator

USENIX Association Student Attendee Grant

The Association will award a limited number of travel and accomodation grants to full-time students interested in attending the Winter USENIX Technical Conference.

Interested full-time students should contact the Association's Executive office (*office@usenix.org*) for an application form soon. Applications must be returned no later than January 3, 1990.

Ellie Young
Executive Director

Call for Papers: Summer 1990 USENIX Conference

Anaheim, California, June 11-15, 1990

USENIX continues to seek papers describing new and interesting work. However, the Summer 1990 Technical Conference also seeks to include papers that emphasize retrospectives, analyses of tradeoffs, and critical thinking about where we are, how we got here, and why we're here. Thus, the theme is:

*Beyond Mere Data:
Perspective, Insight, Understanding.*

Some sessions will follow the normal 3-paper format, with questions following each talk. In other sessions, the speakers will form a panel, following the talks, first to compare approaches, and then to take questions from the audience. In some cases, other experts may be added to the panel to broaden the discussion. Especially desirable are sessions where several important different viewpoints are represented, and proposals for such sessions are welcome.

Appropriate topics include, but are not limited to:

- Software release systems
- User interfaces, windowing, graphics
- Compilers, debuggers, tools, run-time issues
- File systems
- Distributed systems
- UNIX kernel approaches
- Fault-tolerance, reliability, or security
- Computer architectures that stretch UNIX

We will accept full papers, but require at least an abstract and outline, in a form that gives the committee confidence in the final paper. A submission should be 2-3 typewritten pages and include the following:

1. Author names, addresses, telephone numbers and E-mail addresses.
2. Abstract: 100-300 words (half a page) to be included in the final paper.
3. Outline: 1.5-2.5 pages, giving the major headings of the paper, plus a few sentences per section that give the major points that will be covered in that section in the final paper.

The following is a sample outline, which is not necessarily appropriate for all papers, but which illustrates the important topics. The purpose of an outline should be to convince the committee that something interesting and important will be said in the final paper.

1. Introduction

- Background.

Introduce the problem to be solved; why is it important?

Reference previous work; make sure the committee knows the wheel is not being reinvented.

2. How We Solved the Problem

- More details on the problem and its issues.
- Design decisions and tradeoffs, and why they were made.
- Implementation issues.

3. Evaluation

- Data, on performance, effort required.
- How well does it work?
- What would we do differently?
- If it failed, why? and what can we learn from it?

4. Conclusion

- Summarize the paper, emphasizing why it is important, and what was learned.

5. References

- List at least a few key references, preferably to other people's work.

The final paper should retain the 100-300 word abstract, include illustrations (where needed), and citations to relevant literature. Only previously unpublished submissions will be considered, although "retrospective" papers may describe work done years ago. Thinly-disguised product announcements are rarely accepted. Final papers should contain 8-12 pages of single spaced typeset materials. All final papers must be submitted in a camera-ready format or electronic format (*troff-ms* if possible). Typewritten or dot-matrix output is not acceptable. For authors without access to a laser printer or typesetter, appropriate facilities will be provided by the program chair.

Please submit abstracts with outline and proposals for sessions as soon as possible, and mail one hard-copy and one electronic-copy to the addresses below. The final deadline for receipt of submissions is **February 7, 1990**. Abstracts received after this deadline will not be considered. Notification of acceptance or rejection will be made by March 9, 1990. Final camera-ready papers are due by April 17, 1990.

John R. Mashey
Anaheim USENIX Technical Program
MIPS Computer Systems
930 Arques Ave
Sunnyvale, CA 94086
Internet: anaheim@mips.com
UUCP: uunet!mips.com!anaheim
Phone: (408) 991-0253
FAX: (408) 720-9809

Please include your physical and electronic mail address in all correspondence.

Program Committee:

John R. Mashey, (Chair)
MIPS Computer Systems
Clem Cole
Cole Computer Consulting
Doug Comer
Purdue University
Tom Ferrin
Univ. of CA - San Francisco
James Gettys
Digital Equipment Corp.
Lori Grob
Chorus Systems
Douglas P. Kingston III
Morgan Stanley & Co., Inc.
Heinz Lycklama
Interactive Systems Corp.
M. Douglas McIlroy
AT&T Bell Laboratories
Joe Moran
Legato Systems, Inc.
Pat Parseghian
Princeton University
Lawrence Rosler
Hewlett Packard
Bill Shannon
Sun Microsystems, Inc.

Executive Office Staff Changes

The Association has hired Carolyn Carr to be its publications manager. She will coordinate the production of the newsletter and workshop proceedings, as well as provide advice and assistance to the Executive Director on a variety of issues and new projects. Carolyn also owns her own creative services business, which provides graphic design and marketing communications consulting and production services. We should be able to put her expertise to good use, as the Association's activities continue to grow!

Toni Veglia has been hired to replace Eeva McFeely as the new receptionist for the Berkeley office. She will be handling most of your requests, so keep those cards and letters coming.

Call for Papers: USENIX C++ Conference

USENIX is pleased to host its second full C++ conference in San Francisco, California, April 9-11, 1990. We intend this conference to be of interest to a broad range of C++ users and potential users. Even if you have never written a C++ program, you will probably be able to learn enough from the tutorials to follow the technical sessions. This announcement provides early information about the dates of the events as well as persons to contact for further information. The pre-registration packet containing detailed Conference information and hotel reservation information will be mailed in January, 1990.

The meeting headquarters will be the San Francisco Marriott Hotel.

Schedule of Events

Tutorials, April 9

The tutorial program is ideal for people who have been thinking about using C++ but haven't had the opportunity to learn it, as well as experienced users of and researchers in the language.

Please contact the program chair if you are interested in giving a tutorial or have a topic you would particularly like to see covered.

Technical Sessions, April 10-11

The technical sessions will cover the spectrum of work on and with C++, spanning the diversity of its users and applications, and showcasing current research and development. The technical sessions will focus on the current strengths and weaknesses of the language, show where it is and where it is going, and act as a forum for discussion of its future.

Papers are solicited on all aspects of C++, including:

Applications
Libraries
Programming environments
Case studies
New or improved implementations

Extended abstracts (no more than 2 pages) or papers (9-12 pages) must be received, either electronically (preferred) or on paper, by Friday, January 12, 1990. Authors will be notified of acceptance by February 5 and must submit a full paper electronically and in camera-ready form by April 9.

Queries about the technical program and all electronic submissions (*n/troff*, TEX, or PostScript preferred) or camera ready copies should be directed to:

Jim Waldo
CHR 03 DE
Apollo Computer
300 Apollo Drive
Chelmsford, MA 01824

waldo@apollo.com
decvax!apollo!waldo
(last resort) (508) 256-6600, ext. 5747

Program Committee:

Jim Waldo	Apollo Computer, chair
Andy Koenig	AT&T
James Coggins	Univ. of North Carolina Chapel Hill
Martin O'Riordan	Microsoft
Geoff Wyant	Apollo Computer
Roy Campbell	Univ. of Illinois Urbana-Champaign
Peter Canning	Hewlett Packard

EUUG

European UNIX® systems User Group

EUUG CONFERENCES — The World of UNIX at Your Feet



Twice each year — in the Spring and Autumn — the EUUG holds major International Conferences encompassing all the most interesting developments and activities associated with UNIX.

These events are unequalled anywhere in the world for their content and the very high level of speakers — invited from the leading academic and industrial UNIX centres in the USA, Europe and the Far East.

The importance of the Conferences — which are accompanied by tutorials and exhibitions — is underlined by the fact that well over 2000 delegates have attended the last seven events in Florence, Manchester, Finland, Dublin, Lordon, Portugal and Brussels.

If you are at all involved in the field of UNIX, then the EUUG Conferences should not be missed. Indeed, you can make no better start than to put these dates in your diary:

Vienna, Austria	Autumn 1989 (18-22 Sept)
Munich, Germany	Spring 1990 (23-27 April)
Nice, France	Autumn 1990 (22-26 Oct)
Norway	Spring 1991 (20-24 May)
Hungary	Autumn 1991 (16-20 Sept)
Jersey	Spring 1992 (Dates to be announced)

or call the EUUG Secretariat for further details:

European UNIX® systems User Group

Owles Hall, Buntingford, Hertfordshire SG9 9PL,
UK

Tel: + 44 763 73039 Fax: + 44 763 73255

Network address: euug@inset.uucp

Call for Papers: AUUG Conference and Exhibition 1990

Melbourne, Australia, September 25-28, 1990

The 1990 Conference and Exhibition of the Australian UNIX systems User Group will be held at the World Congress Centre in Melbourne. Tutorial sessions will be held on the 25th and the conference proper from the 26th to the 28th of September 1990. The conference theme is:

UNIX: the Computing Platform for the 90s

Papers are invited on topics which will interest an audience of either Research, Technical, Industry, or Commercial UNIX users. Some suggested topics are:

Future Directions	Standards
Networking	Security
Project Management	Productivity Tools
Database	System Administration
User Interfaces	Windowing Systems
Real Time Systems	Multiprocessing

Papers that describe current Work in Progress, and papers on other topics relevant to the UNIX user community are also welcome.

Authors of each paper accepted will receive ONE complimentary admission to the conference and the dinner.

AUUG will again hold a competition for the best paper by a full time student at an Australian educational institution. The prize will be an expense paid return trip from within Australia to the conference to present the winning paper. A cash prize in lieu of this may be made at the discretion of AUUG. Students should indicate with their abstract whether they wish to enter the competition. AUUG reserves the right to not award the prize if no entries of a suitable standard are forthcoming.

A special issue of the group's newsletter AUUGN containing the conference proceedings will be printed for distribution to the attendees at the conference and mailed to AUUG members who do not attend.

A 1000-2000 word extended abstract is required which describes the nature of the paper and a summary of conclusions and/or results.

Acceptance of papers will be based on the abstract and will be subject to receipt of the final paper by the due date. The Programme Committee Chair reserves the right to withhold final acceptance until the final paper is received. Abstracts and final papers should be submitted to:

John Carey
AUUG 90 Programme Committee Chair
Labtam Information Systems Pty. Ltd.
43 Malcolm Road
Braeside Victoria 3195 AUSTRALIA

Phone: +61 3 587 1444
Fax: +61 3 580 5581
Telex: LABTAM AA335500
Internet: john@labtam.oz.au
UUCP: uunet!munnari!labtam.oz!john

Final Papers should contain a 100-300 word abstract and 10-20 pages of 10 point single spaced text.

Important Dates

Receipt of Abstracts	5 Feb. 1990
Letters of Acceptance	5 Mar. 1990
Receipt of Final Papers	6 Aug. 1990
Tutorials & Conference	25-28 Sep. 1990

People wishing to present tutorials should contact:

Chris Maltby
AUUG 90 Tutorials
Softway Pty. Ltd.
79 Myrtle Street
Strawberry Hills NSW 2012 AUSTRALIA

Phone: +61 2 698 2322

All enquiries regarding registration, accommodation, and the Exhibition:

AUUG 90 Secretariat
c/o ACMS
26 Hopewell Street
Paddington NSW 2021 AUSTRALIA
Phone: +61 2 332 4622
Fax: +61 2 332 4066

Long-Term Calendar of UNIX Events[†]

1989 Dec 5-6	JUS UNIX Fair '89	Tokyo, Japan
1989 Dec 6-8	Sun Users Group Conf	Anaheim, CA
1989 Dec 8-9	UNIX Asia '89	Sinix; Singapore
1989 Dec 11-13	UKUUG	Cardiff, Wales, UK
1989 Dec 11-15	OSI Implementors Workshop	NIST; Gaithersburg, MD
1990 Jan 8-12	IEEE 1003	New Orleans, LA
1990 Jan 9-10	UNIX in Government	Ottawa, Ont.
1990 Jan 20-26	DECUS	Toronto, Ont.
1990 Jan 22-26	USENIX	Omni Shoreham, Washington, DC
1990 Jan 23-26	UniForum	Washington Convention Ctr., Washington, DC
1990 Feb 14	UKUUG Sys. Admin. Workshop	Inst. of Ed, London, UK
1990 Mar 5-6	X3J11	New York, NY
1990 Mar 26-29	DECUS	Vasteras, Sweden
1990 Mar 26-30	AFUU	Paris, France
1990 Apr 9	POSIX APP Workshop	NIST; Gaithersburg, MD
1990 Apr 9-11	USENIX C++ Conference	San Francisco, CA
1990 Apr 23-27	EUUG	Munich, Germany
1990 Apr 23-27	IEEE 1003	Salt Lake City, UT
1990 May 7-11	DECUS	New Orleans, LA
1990 May 30-Jun 1	UNIX/90	/usr/group/cdn; Toronto, Ont.
1990 Jun 11-15	USENIX	Marriott Hotel, Anaheim, CA
1990 Jun 11-15	ISO WG15 (POSIX)	Paris, France
1990 Jul 9-11	15th JUS Symposium	Toyko, Japan
1990 Jul 11-13	UKUUG	London, UK
1990 Jul 16-20	IEEE 1003	Danvers, MA
1990 August	* Security	Portland, OR
1990 Autumn	* Mach	
1990 Sep 25-28	AUUG	World Congress Centre, Melbourne, Australia
1990 Oct 22-26	EUUG	Nice, France
1990 Oct 31-Nov 1	UNIX Expo	New York, NY
1990 Nov 5-9	Computer Communication Conf.	ICCC; New Delhi, India
1990 Nov 15	POSIX APP Workshop	NIST; Gaithersburg, MD
1990 Nov 15-16	16th JUS Symposium	Osaka, Japan
1990 Dec 4-5	JUS UNIX Fair '90	Tokyo, Japan
1990 Dec 10-14	DECUS	Las Vegas, NV
1991 Jan 21-25	USENIX	Grand Kempinski, Dallas, TX
1991 Jan 22-25	UniForum	Infomart, Dallas, TX
1991 Feb	UNIX in Government	Ottawa, Ont.
1991 Feb 18-22	DECUS	Ottawa, Ont.
1991 May	UNIX 8x/etc	/usr/group/cdn; Toronto, Ont.
1991 May 6-10	DECUS	Atlanta, GA
1991 May 20-24	EUUG	Tromso, Norway
1991 Jun 10-14	USENIX	Opryland, Nashville, TN
1991 Sep 16-20	EUUG	Budapest, Hungary
1991 Dec 9-13	DECUS	Anaheim, CA
1992 Jan 20-24	USENIX	Hilton Square, San Francisco, CA
1992 Jan 21-24	UniForum	Moscone Center, San Francisco, CA
1992 Spring	EUUG	Jersey, UK
1992 May 4-8	DECUS	Atlanta, GA
1992 Jun 8-12	USENIX	Marriott, San Antonio, TX
1992 Autumn	EUUG	Amsterdam, Netherlands

Compiled with the assistance of Alain Williams of the EUUG and Susanne Smith of Windsound Consulting.

* USENIX Workshops

USENIX Board Studies UUCP

At the recent USENIX board meeting in Vienna, USENIX and EUUG agreed to jointly study UUCP, and I have agreed to be the contact and collection point for thoughts, proposals, suggestions, and flames.

Most people would agree that UUCP has many problems. Compatible versions are not available throughout the entire UNIX community, and its penetration of non-UNIX systems is minimal. Maintaining and administering UUCP threatens the sanity of even reasonably stable individuals, and is seriously damaging to UNIX hackers. The robustness and performance of the transmission protocols is open to question. The CPU and disk load that UUCP places on the operating system can and probably should be improved. ISO and X.25 compatibility are of interest to the Europeans. The list goes on.

So what can USENIX do about this? As you recall, a similar series of discussions about Usenet led to sponsorship of the Stargate experiments and eventually establishing and spinning off the very successful UUNET service. Some of the concrete actions that we have discussed are:

- Sponsoring a public-domain re-implementation of UUCP.
- Picking up and distributing one of the existing re-implementations.
- Hiring people to make studies or specific proposals.

As Treasurer of USENIX, I naturally objected to the third of these alternatives, which is why I got stuck with doing it.

In my view, there are several things that a YACP (Yet Another Communication Protocol) program should do:

- Be able to send and receive from existing UUCP sites.
- Be sensitive to the security risks of network communication.
- Be written for today's machine memories, disks, and network traffic.

- Talk at least a few other protocols; ideally, make it easy to add new protocols through streams or dynamic linking.
- Allow administration of incoming and outgoing traffic that is both easy and helpful for the naive, and not sadistic to the full-time administrator.
- Be widely available, even for non-UNIX licensees, through some form of flexible licensing scheme.
- Be robust enough that the hackings of cretins not disrupt the network, and produce clear error messages.

From the organizational point of view, there are also some non-technical questions:

- What should we do, in detail? Can we do the work in stages?
- When we decide what to do, who does it?
- How much does it cost? How do we pay for it?
- How do we distribute the final product? On what terms?
- If distributed in source form, how do we keep people from "improving" it into incompatibility or worse?
- Is this really the way we should be spending our money?

USENIX is fortunate to have significant financial reserves, and can afford to do this project right, if we decide to do it at all. That is where you come in. We would like to hear from our members on all aspects of this project - technical, organizational, the works. Alternative projects are also gratefully accepted. Please send mail to:

scj@usenix.org

We will be discussing this project at the next board meeting in January, and hope to decide then how (or whether) to move forward.

Steve Johnson

Summary of Board of Directors' Meeting

Vienna, Austria, September 17 and 19, 1989

Attendance: Stephen Johnson, Marshall Kirk McKusick, Sharon Murrel, Michael O'Dell, Alan Nemeth, John Quarterman, Deborah Scherrer, Ellie Young, John Donnelly, Daniel Klein, Judith DesHarnais, Ernst Janich, Johan Helsingius, Rick Adams, Alain Williams, Philip Peake, Neil Todd.

Workshops

Systems Administration III. Donnelly reported that the dual track format of the tutorials was well received. Attendance figures: 219 total; 183 for tutorials; 199 technical sessions only; 20 tutorials only.

Distributed Systems. O'Dell said there were good papers, and this topic might become a recurring workshop.

C++ '90 Conference. Young reported that a program committee was formed, and the Call for Papers had been mailed.

Security '90. Nemeth would contact Matt Bishop with recommendations for format guidelines, and he volunteered to serve as board liaison.

Software Development Environments Workshop. Nemeth reported that he had received a favorable reply from the technical head of the SIGMA Project in Japan, regarding co-sponsorship of an international workshop to be held in the Fall of 1990.

D.C. '90 Conference

Klein reported that 83 submissions were received and 27 papers had been accepted. He felt that extended abstracts worked very well. A panel on ethics in the computer industry would be offered.

Quarterman wanted to advertise daycare as available in D.C. It was decided to allocate \$5,000 to offer and possibly underwrite this service to interested participants.

Abolish Winter Conferences

Quarterman said he put this item on the agenda because it was perceived that there was a problem of a weak technical program at the recent conferences. Nemeth said that perhaps

we've really switched to having ten conferences per year (e.g., workshops have increased). O'Dell felt that the paper quality isn't the issue, that the problem is one of drawing conclusions, and our need for strategic planning as an organization. Nemeth stated that we need risk profiles in order to ascertain loss of income and possible penalties. A subcommittee of Johnson, O'Dell, and Nemeth was formed to discuss this issue.

Professional Development Seminars

Donnelly stated that the first one would be held in Chicago and consist of three sessions. He was optimistic about enrollment.

Standards

Quarterman reported on the activity of the ISO/IEC JTC1 SC22 WG15. Johnson wanted to know why we are supporting this activity. Quarterman replied that it is an attempt to prevent standards from prohibiting innovation.

Vendor Sales at Conferences

Young went over our attorney's findings that the general policy prohibiting vendors from making sales at the conferences is unnecessary from a tax perspective, and asked if there would be another reason for prohibiting sales on the floor. It was decided to allow selling on the floor, and that Donnelly should continue to screen vendors and be the regulator of taste.

Executive Office Report

It was suggested that we post on the net the dates of the upcoming board meetings with a set of topics, and suggest that members contact board members with input.

Budget – Revised Projections for 1989

Young went over the budget which provided an overview of the current finances as of July 1989, as well as projections for where we'll stand in November. In most expense categories we had realized savings. The board had also earmarked \$162,500 in discretionary funds during this fiscal year.

Proposed Budget for 1990

The board went over Young's list of assumptions in preparing the budget for next year. This led to a discussion about making adjustments in compensation, fees, and offering discounts. The speakers' compensation committee would meet again, and Young would prepare a proposal for discounts. It was also decided to leave the projected net change figure for next year in place subject to getting a risk profile. The budget was approved.

EUUG Relations

Johan Helsingius ran through the structure of the EUUG boards. Their governing board consists of two representatives, who are board members of a national group in each country (approximately 20 countries). They give direction to the executive committee, and strategic/overall planning for the EUUG as a whole. The executive group is a subcommittee of the governing board that handles the day-to-day matters and is self-elected.

EUUG Executive Board:

Neil Todd (events/tutorials)
Ernst Janich (events)
Kim-Biel Nielsen (pr)
Teus Hagen (co-chair)
Michel Gien (co-chair)
Nigel Martin (finances)
Daniel Karrenberg (networks)
Philip Peake (publications)

Co-opted members (on trial):

Johan Helsingius
Norman Hall
Francis Brower

Conference Chair Proposal

Quartermann's proposal for a model for assignment of conference chairs was approved, as follows:

1. Assign no chair to any conference without a specific item for that purpose on the agenda of the board meeting.
2. Place assignment of the chair for any conference without a chair on the agenda of the board meeting 18 months before the conference.
3. State reasons in the minutes of the board meeting when assigning a chair to a conference more than 18 months in advance.

4. Require anyone who asks to chair a conference to submit a written proposal, and assign a chair to a conference without a written proposal only in exceptional cases with reasons stated in the minutes.

5. Post (in *:login:*, on Usenet, and in posters) a request for proposals to chair a conference to coincide with the conference 24 months in advance of any conference that needs a chair.

Quartermann provided a sample proposal with various points.

It was also agreed that while the board liaison must be an actual member of the Board at the time that the proposal is accepted, and the chair appointed, he/she may continue to be liaison even after retiring from the board.

Joint Workshop – EUUG and USENIX

The EUUG representatives expressed their desire to hold a joint workshop with USENIX in Europe at a location without a national group, some time in the near future. The primary goal would be to get technical developers together to exchange ideas and bring people in that are more leading edge. It was agreed to extend to EUUG an expression of interest in a joint workshop, the topic, date and location to be established by joint subcommittees of each group, and that we allocate up to \$10,000 to be expended in matching funds with EUUG in the planning and preparation for this event. Any profit or loss will be split between each group. Nominal time frame would be Fall of 1990.

Public Domain UUCP Implementation

It was decided to allocate \$2,500 to pay for the cost of generating a full proposal for the implementation, management, and production of a public domain version of UUCP, and that USENIX would proceed unilaterally, but would be willing to work jointly with EUUG. Johnson volunteered to collect information.

Next Board Meeting

It will be held at the Omni Shoreham Hotel in Washington, D.C., on January 21, 1990, and continuing on January 22.

EY

Audio I/O with the NeXT Computer

Michael Hawley

NeXT Inc. / MIT Media Laboratory

ABSTRACT: The NeXT machine is the first widely available computer with a built-in microphone. It is the first with a DSP, and with high-quality audio output. As such, it helps to usher in the great age of audio-rich computing, something like the precedent set by Al Jolson for film. Like movies, applications of sound in computing will not be limited to crude "talkie" interfaces, but will grow to include sound design of all kinds. The fact that these resources are available at the lowest common denominator means that applications can be written which can rely on reasonable digital audio facilities. In this paper we will outline some of the system tools for working with audio – the Sound Kit, the Music Kit, and related code – and discuss some audio-intensive applications which are emerging.

Introduction: *Al Jolson is to NeXT as THX is to ...?*

Computing is at last moving out of the silent era and into the great age of "talkies." Glancing back at the history of cinematic technology, our work in inventing audio-rich computers today seems just as balkan as the skirmishing that went on from 1900 to 1930. In 1895, Edison introduced the *Kinetophone*, which supplied musical accompaniment for a "peep show." There was no synchronization. It flopped. Shortly after that, Leon Gaumont presented the *Chronophone* in France in 1902. The Chronophone played sync sound and picture, and in a smart entrepreneurial move, Gaumont filmed vaudeville acts as the material to bootstrap his invention. The Motion Picture Patents Company licensed the technology, but Chronophone failed because the system was expensive, insufficiently amplified, produced coarse sounds, and drifted out of sync. This was about 1913, and Edison and Gaumont were only two of dozens (*Cameraphone*, *Vivaphone*, *Synchroscope*, ...). All tried to mate the silent movie to the phonograph, and uniformly failed because of combinations of cost, amplification, bad synchronization, and lack of quality.

In 1913 Edison announced the Kinetophone again. He claimed to have solved the talkies problem. He used a giant phonograph for maximum amplification, and belts and pulleys between the projection booth and the stage to sync the phonograph with the projector: some current attempts to integrate audio

in computing are not unlike this! Again the technology proved inadequate: during performances, the sound slipped out as much as 10 or 12 seconds; audiences booed the picture off the screen. Contracts were rejected, Edison's factory in West Orange burned to the ground, and that was that. In the early 20's, *Phonofilm* was invented by Lee DeForest, who also had patented the audion amplifier tube in 1907. Phonofilm was a major advance – voice was recorded *onto the film*, in sync – but DeForest was a lackluster entrepreneur, and failed to secure the key deals and patents required.

Eventually, of course, it was AT&T that succeeded, through its daughter company Western Electric, and contracts with Harry Warner and sundry other Warner Brothers. Experiments began in 1925, and flourished with the formation of *Vitaphone*, a Warner subsidiary. By 1927, Vitaphone premiered *The Jazz Singer* starring Al Jolson, and a number of other films – the first true talkies. Even though *The Jazz Singer* got lukewarm reviews, Jolson's songs became hits in their own right (and Jolson instantly signed a \$100,000 contract for three more movies).

The point of all this is that the invention of a successful recording and reproduction system for sound in movies took place over a span of three decades and with considerable skirmishing – and that was only the pioneering work that led to the earliest talking films. There is much more to sound in movies than just speech, though, and the art form and technology have been evolving steadily since

then. Jack Foley is remembered as the developer of "Foley effects" in the 1930s – human non-vocalic sound effects, like footsteps, nose crunches, eating noises – and with Star Wars, Ben Burtt launched the field of "sound design." Film music has also evolved into its own genre, employing a huge industry of musicians and composers; theater sound systems have evolved in fifty years from *Vitaphone* to *THX*.

With this in mind, think about computing, and the noises made by most computers compared to the potential experience of an audio-rich computer. We are at a point now where technology that can support decent audio processing in a general and widely used computing system is becoming available. The lesson from the past is not that we should let inventors slug it out, and wait for AT&T to solve the sound problem like they did in 1925, but rather that there are compelling reasons why general-purpose audio processing of the highest quality should be made a kernel element of computer systems. NeXT, of course, is the first ambitious example, but not the last. It is important to keep this in mind because sound will contribute enormously in shaping the personality of machines to come. What follows is an overview of the facilities packaged with the NeXT computer, hard and soft.

Overview

NeXT provides sufficient hardware and software for a wide variety of basic audio applications, from voice mail to speech synthesis, speech recognition, or sound effects design in the user interface. The software available for sound processing is C or Objective-C (an object-oriented dialect of C), and the MACH operating system provides considerable support in low-overhead scheduling and driver code.

Hardware

Voice-Quality Input

The NeXT has a bundled microphone (to be mounted in the bezel at the bottom of the monitor) and a high-impedance microphone jack. These feed into a CODEC a/d converter. The CODEC part has an anti-aliasing prefilter

and generates 8012.8Hz 8-bit mu-law coded input – that is, about 8,000 bytes per second for telephone quality speech input. The mu-law coding provides a 12-bit effective dynamic range compressed to 8 bits. I/O is interfaced through DMA implemented in a custom gate array.

High-Quality Sound Output

The stereo D/A converter operates at 44.1KHz in each channel with 16-bit linear quantization, just like a commercial CD player. A 1KHz maximum-amplitude sinusoid played through the DAC generates a 2V RMS signal at the audio jack. The converter includes de-glitching and anti-aliasing filters. A speaker is built into the base of the monitor and provides surprisingly good sound. Additionally, stereo headphone (mini) jacks and a pair of gold-plated RCA stereo audio jacks are accessible in the back of the monitor for high-fidelity. Cheap sound (i.e., 8KHz or anything else lower than 44.1KHz) is of course interpolated up to 44.1KHz to feed the converters.

High-Quality Sound Input

There is none. With present technology (and cost) it does not make sense to force this into the lowest-common-denominator machine. However, it is easy to feed high quality data directly into the DSP port. There are already relatively inexpensive third-party products (around \$600) that make it easy to flow analog and digital audio at high sampling rates directly into the machine.

DSP

The digital signal processor comprises a Motorola DSP56001 running at 25MHz; memory-mapped DMA access at 5Mbytes/second to the host interface; 8K 24-bit words of zero-wait-state RAM (local to the DSP); and a D-15 connector providing access to the DSP's SSI and SSC ports.

The DSP executes 12.5 million instructions per second and, in a single instruction, can perform a 24x24 bit multiply, a 48+56 bit addition, two parallel data moves, an instruction fetch, and two index updates. 24-bit data paths are well suited to high-quality audio processing. The DMA access (the interface

between host processor and DSP) provides access to the eight byte registers of the DSP. For more information about the DSP chip, read the documentation from NeXT or Motorola.

Software

Overview

For the purposes of this discussion, software can be divided into four main components: DSP-related software (e.g., driver and other direct DSP support), the Sound Kit, the Music Kit, and application-level code.

Sound Kit

The Sound Kit is a library for accessing basic sound capabilities in the NeXT computer. Like the other major software "Kits," the Sound Kit is object-oriented, which, among other things, facilitates the handling of various formats of sound. Casual use of sound is easy:

```
id nyuk = [Sound newFromSoundfile:  
           "'ThreeStooges.snd''];  
[nyuk play];
```

The Sound Kit also makes possible detailed, nitty-gritty access to sound in all its formats. It manages playing, recording, reading, writing, copying, etc., and makes much use of operating system primitives for virtual memory management, interprocess communication, and thread-level scheduling to efficiently process sound. For example, to record sound into a Sound object, send the object a *record* message. When the message is received, a thread (a lightweight process) is activated to fetch and store the samples, typically reading from the CODEC. This typically happens asynchronously, so that the calling process can continue doing other things (like display management, as when implementing bouncing VU meters). When sound input finishes, a message can be sent to the parent in a similar way.

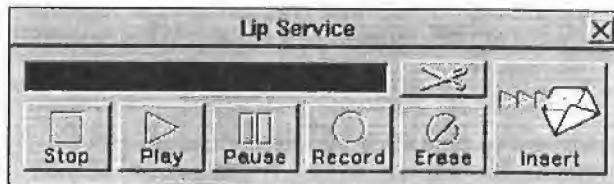
Formats

A variety of formats are supported, from low-quality to high, mono, stereo, etc. The Sound object uses the DSP for run-time format

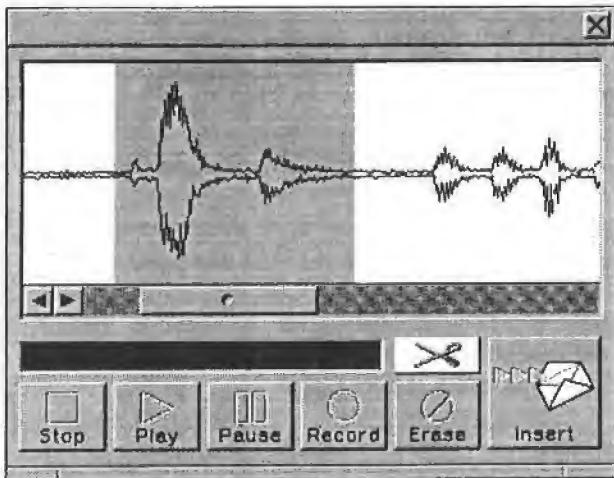
conversion of sampled sounds, which takes some of the load off the main CPU. The Sound Kit also supports DSP sound synthesis instructions – sounds which are described not by lists of samples, but by DSP algorithms and data streams. In any event, [sound play] and similar Kit routines work transparently. In theory, sounds may be multi-channel, but in practice the processor and disk bandwidths won't sustain more than about two channels of 44.1KHz stereo. (In fact, the optical disk does not write sufficiently fast to permit stereo recording in realtime at this rate; Ethernet barely sustains speech).

Views

The SoundView Class provides some display facilities that are compatible with the rest of the NeXT user interface conventions. The SoundView can draw, scale, select, scroll, etc. Sound is, at the moment, typically displayed as a waveform or amplitude trace, but other display methods can easily be applied. The Application Kit and Interface Builder (user interface construction tools) make it possible to stitch together sounds, views, and other interface objects easily. Associating an arbitrary sound effect to a button click (say) is simply a matter of dragging a sound file onto the button. These are the building blocks that are the foundation for other applications. For instance, the NeXT Mail program supports a simple form of voice mail, which looks like this:



The horizontal black bar holds a peak meter which bounces when you speak. Pressing the scissors button flips open an editable SoundView, which lets you scroll and edit the sound, as shown in the illustration on the next page.



Music Kit

The Music Kit provides library access for building music applications. Support for music representation, performance, and DSP-based synthesis and processing are all available. The general design emphasis integrates the gesture-level of control (e.g., MIDI and similar control-level encodings) with the low-level timbral control made possible by academic sound synthesis systems (like MUSIC-5), and cultivate it all in a rich application system. Computer music is a fruitful application area since it demands not only a mixture of technology, art, and aesthetics, but also (unlike most speech work, say) really pushes issues of attention and beauty. The speech community has had to invent a special pigeonhole for research to make digital speech captivating – prosody – and it is too often neglected in general. In music, a lousy-sounding piece may be either a failure, or deliberate, but in any event, the question of producing compelling or evocative sound is central. Moreover, the demands of musical research are typically not as narrow as those of speech. Together with sound effects, mixing, and processing, these are the main streams of flow required for audio rich computer of cinematic quality.

Like the Sound Kit, the Music Kit controls instrument generators in the DSP, but in a way that is more general than commercially

packaged music synthesizers. Music is represented as a hierarchy of Score, Part, and Note objects. We will not discuss the Music Kit's elements in deep detail (one can read the copious NeXT documentation, or *Sound and Music on the NeXT Computer*, by Smith, Jaffe, and Boynton, AES 1989). What is of interest here, though, is the fact that the Music Kit manages general-purpose code for controlling the DSP. *Unit Generators* and *Synth Data* elements are the basic algorithmic building blocks for audio networks. They are typically expressed as little algorithms of calls to 56000 assembly code macros. *Synth Patches* are networks of these; and *Synth Instruments* are renderers that play notes by assigning them to instances of *Synth Patches* – this is to say, there is considerable software support for writing, loading, and scheduling networks of signal processing elements. Arching over all of this is an *Orchestra* class which oversees all the instrument processing done in the DSP. Given a general setup such as this, it is easy to exceed the realtime bandwidth of the DSP, so the Music Kit makes it possible to generate compute-intensive sound files out of realtime when necessary, without loss of generality.

DSP Software

The DSP software presently falls into two categories – Music Kit support and array processing support (and consequently, driver-level code for setting up the DSP to manage computation like this). Over time, specialized support for speech, signal processing, etc, will certainly evolve. The Monitor for the Music Kit implements things like DMA support, buffering of sound, unit generators (which are DSP programs) and other things needed by the music kit. Unit generators include components like adders, multipliers, allpass filters, basic oscillators, delays, etc. The array processing software includes various vector and array function macros, like FFTs, digital filters, etc. A program called *dspwrap* translates a DSP macro to a C callable function (that is, the host program is given a hook to call corresponding code in the DSP). There is a substantial body of code for supporting host/DSP communication via interrupts, messages, FIFOs, etc.

Conclusions

At the moment, most computers are like silent movies, and the audio channel is virtually unused. There is a *PostScript* for graphics, but not for sound. NeXT is the first computer to provide a facility for fairly general-purpose sound I/O, and even before the 1.0 release, has already shown applications like voice mail, CD-quality storage and playback, speech recognition (the *Sphinx* project at CMU has been ported to the NeXT machine; the printer can talk when it runs out of paper), sound effects in the interface (e.g., physical simulations of Billiards or Cessna flights including sound effects), real-time FFT and scope displays, etc. Certainly over the coming year or two, the NeXT will begin to recognize its owner's voice (or gender), and respond to simple spoken menu commands, but uses of audio in computers go far beyond simple speech processing and will eventually recapitulate many of the developments in cinema. In ten or twenty years, we think using a computer without sound will be like experiencing *Star Wars* without a soundtrack, so computer systems need to be designed with appropriate generality in mind.

Acknowledgements

Julius Smith, Gregg Kellogg, David Jaffe, Dana Massie, and Lee Boynton are the chief architects of the sound, music, and DSP code. Michael McNabb and Roger Dannenberg provided numerous features and design insights in the Music Kit; Doug Keislard, Doug Fulton and Richard Crandall created documentation and demos of the audio software. It was Bud Tribble who sparked NeXT into putting the DSP56001 in every machine, and finally, Steve Jobs who had the sense of vision and commitment to make general-purpose high-quality audio a key building block in the NeXT computer.

References

More information on the specifics of NeXT audio software can be found in *Sound and Music on the NeXT Computer*, by Smith, Jaffe, and Boynton, AES 1989, and in the documentation provided by NeXT. Motorola provides documentation for the DSP. One of many good books on sound in movies is *Film Sound: Theory and Practice*, edited by Elisabeth Weis and John Belton, Columbia University Press, 1985.

Book Review:

!%@:: A Directory of Electronic Mail Addressing and Networks

by Donnalyn Frey and Rick Adams

(\$26.95; O'Reilly and Associates, Sebastopol, CA, 1989)

Reviewed by Peter H. Salus

Open Software Foundation

How many times each day does one get an email message bounced?

Where can one look for information on the myriad electronic mail networks around the world?

Is there a way to stop your postmaster from going mad?

The answer to the first question may not be accessible in this ungainly, invaluable book; Frey and Adams have (in just under 300 pages) answered the others. If you don't want to read this review, here's the bottom line: If you use electronic mail outside of your own site, **buy this book**. It will redeem its cost in but a few days.

In fact, together with J. S. Quarterman's *The Matrix*, which is complementary to Frey and Adams, !%@: will yield a genuine understanding of both the ways in which email works and the links among the global networks.

Frey and Adams [F&A] – not exactly strangers to the UNIX or the USENIX communities – have organized their handbook in a sensible way:

Chapter 1, "A User Introduction to Electronic Mail," serves as a first-rate tutorial to message formats and addressing. If you ever wanted to understand the differences between @ and % or @ and !, here you are. F&A not only explicate the various addressing formats, they expound clearly and concisely on the nature of local names, mailboxes, and domains. They even manage (p. 11) to be light-hearted about the British (and New Zealand) peculiarity of writing their mail addresses "backwards." There are a few overly friendly footnotes, e.g. the explanation of the "happy-face," but these are bearable.

Chapter 2, "Networks," comprises over two-thirds of the book (pp. 23-231). It lists (in alphabetical order) all the nets I've ever heard of – and a number I'd never heard of before. [Actually, ATTMail is missing. When I asked Frey about this, I was told that they had never responded to (repeated) requests for information. *Tant pis.*] If you need information on NorthWestNet – the Northwestern States Network, with nodes in Alaska, Idaho, North Dakota, Oregon, and Washington ("a ring network with a satellite link to the Alaska site ... [maintaining] a satellite link from Oregon State University to NCAR in Boulder, Colorado, USA"), or on ILAN – the Israeli Academic Network – where the contact person is "Avi Cohen, Director, InterUniversity Computer Center, Tel Aviv University ...", it's here. So far as I can tell, the information is accurate. It is concisely presented and there are maps to go with every network. Oh boy! There are misprints, but a month's use of F&A disclosed none that wasn't self-correcting.

Appendices. There are five appendices and a glossary of terms. Second level domains and ISO codes are covered in Appendices A through D (pp. 233-265); Internet Address handling is covered in Appendix E (pp. 265-268). I suppose that one should quibble with definitions and explanations in the glossary, I find it hard to do so.

The volume concludes with two indices: by name or type of network and by notation [=abbreviation] of network.

!%@: is useful, well-organized and complete. F&A have done the entire user community a tremendous service in producing this volume; Tim O'Reilly deserves a vote of thanks for publishing and distributing it. Buy one today!

Report to EUUG and USENIX on ISO/IEC JTC1/SC22/WG15 (POSIX) Meeting

October 11-13, 1989

Dominic Dunlop

The Standard Answer Ltd.

Introduction

Working Group 15 of Subcommittee 22 of Joint Technical Committee 1 of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC22/WG15) met in Brussels, Belgium, from October 11-13 in order to further the POSIX standardization effort. I was present at the meeting as an observer with the brief of reporting back to you. This report is the second jointly commissioned by the European UNIX systems User Group (EUUG) and USENIX. If you have any comments, or need clarification or further information, please contact me at the mail address above.

First, a summary of the most important aspects of the meeting.

Summary

- The big news is that the working group has recommended that ISO accepts the POSIX operating system interface in its current form as international standard (IS) 9945-1. Assuming that this recommendation is accepted, an international standard which is identical to IEEE Std 1003.1-1988 should be registered by ISO in the next few months.

- During the balloting of the standard at the international level, a number of comments were raised. These will be addressed by the production of a revised International Standard on short order – by next June according to the current schedule. The result will be a version of POSIX in which known problems are fixed, but which is not extended in any way.

- Extensions, such as real-time facilities, transparent network file access, and security features will be added in future releases of the international standard.

- The cooperation of the IEEE POSIX project in producing standards which are acceptable to ISO and to its members is critical to the timely production of ISO standards. Steps were taken to make sure that IEEE documents are produced in a format that is acceptable to ISO, and that IEEE work on the revision of its 1003.1 standard is synchronized with the work of the ISO working group.

- Draft 9 of IEEE 1003.2, the proposed IEEE shell and utilities standard, has been accepted as Draft Proposal (DP) 9945-2. This means that the movement towards an international standard in this area is now officially under way.

- The problems raised by the suggested adoption of the whole of issue 3 of the X/Open Portability Guide as a European prestandard (see report on May, 1989 meeting) seem to have receded: European alignment with a number of formal international standards is finding acceptance as a viable and more useful alternative.

- The working group has set up “rapporteur groups” on conformance testing, internationalization, and security in order to ensure that future international standards for POSIX take account of the developments in, and of the requirements of, these important areas.

- The next meeting of the working group does not take place until June, 1990. Making a virtue of necessity, the group hopes to achieve much before that time.

POSIX as an International Standard

The international ballot period for Draft International Standard (DIS) 9945, *Portable operating system interface for computer environments*, closed at the beginning of September. The DIS is identical to draft 13 of the

IEEE 1003.1 POSIX standard, which in turn is identical, except in details of layout, to Std 1003.1-1988 published by the IEEE.

Of 26 national standards bodies entitled to vote, 19 approved the standard, one (South Africa) abstained, and one (Japan) voted against. (The five remaining countries did not vote.) Broadly speaking, ISO rules require only 75% of those voting to vote in favor in order that a standard is accepted. Where there are only one or two votes against, as in this case, the situation is even more clear-cut. Nevertheless, ISO rules require the technical committee responsible for the standard to show that it has considered the concerns of the objectors, even if it has decided not to address them by amending the draft standard.

Japan's major worry was simply that the document did not *look* like an International Standard – a matter on which France, despite voting in favor, and ISO's Central Secretariat, had also voiced concern. Instead, DIS 9945 looks like what it is – the draft of an IEEE standard – and may consequently be difficult to navigate for those used to ISO's standard format for standards.

This editorial issue could be handled simply by instructing ISO's Central Secretariat to re-enter the document text, and set it in the required format. This would take perhaps a year, and would not address the large number of "non-normative" changes already known to be required in the document as a result of work done by the IEEE over the past year. These changes are currently under discussion within the IEEE as P1003.1a. They are thought not to affect substance of the standard, merely clarifying it, fixing a number of small errors, and adding standard C function prototypes. However, ISO procedures sensibly require that any change to a draft standard must result in a new vote on the amended document, and consequently a further delay to the acceptance of a final standard.

Judging that it was more important to get a POSIX standard out in the field as soon as possible, rather than to ensure that its format and content was perfect in every way, the working group decided on a two step process:

1. Recommend that DIS 9945 is accepted in its current form as IS 9945-1. (The request to split the POSIX standard into multiple documents came as the draft standard was being balloted, with the result that its number has sprouted a -1.) ISO may decide to reprint the existing document, adding cover material to say that it is a standard. Alternatively, the standard may be published as a reference document: a few pages which tell the reader to go and look at a particular ANSI standard. (There is a precedent for this: the International Standards for COBOL and PL/I simply point to ANSI documents.)

If ISO accepts the recommendation, POSIX should become an International Standard within the next six months.

ISO may turn down the request if it judges that the working group's plans to resolve outstanding issues are inadequate. Hopefully, this will not happen, because:

2. The working group has undertaken to produce and ballot an amendment to the standard by 1st June, 1990. The amendment – actually 1003.1a produced by the IEEE – will fix all issues raised during the balloting of DIS 9945. What is more, the working group – or rather, the hard-pressed editor for the IEEE's POSIX project – will merge the addendum with the existing standard, producing a single document in a format acceptable to ISO. This, it is hoped, will be published as a revised standard late next year.

The Future of International Standards for POSIX

In my last report, I noted that the working group had requested that its project be split into several parts, resulting in several standards, numbered 9945-1, 9945-2 and so on, rather than a single standard 9945. This has happened, with the result that the operating system interface will be covered by 9945-1; shell and utilities by 9945-2, and system administration by 9945-3. No other numbers have yet been allocated. It is important to note that the apparent one-for-one correspondence between 1003.1 and 9945-1 will grow more tenuous as time goes on: facilities for real-time processing (1003.4), security control

(1003.6) and transparent file access (1003.8) will be added to future versions of 9945-1. While 9945-2 corresponds to 1003.2, there is no connection between 1003.3 (Test Methods) and 9945-3. Instead, 9945-3 – when it gets off the ground – will be based on the IEEE's 1003.7 work.

I also mentioned last time that ISO standards are supposed to be independent of any particular computer language. 9945-1 will probably lose its ties to C with its second amendment (that is, the amendment after the one described in the previous section). This will introduce a need for a new standard to describe its C bindings, and further standards to describe bindings for Ada, FORTRAN, and so on. While the IEEE language bindings are part of the 1003 project (1003.5 for Ada, and 1003.9 for FORTRAN), ISO practice is to allocate a completely new standard number for bindings work. Consequently, a request for a new number, with three designated parts, has been made. We will not know this number until next June.

Table 1 summarizes correspondence between ISO and IEEE standards.

A word about windowing is in order. Work in a number of JCT1 SCs nibbles at the edges of the issue:

SC2 (Code sets):

Encoding of pictures. There is no connection between this work and X's bitmap distribution format.

SC18 (Office systems):

Office system user interface; Font and character information interchange (lots of this); page layout and document structure (even more of this).

SC22 (Languages):

Form interface management system – a new project involving interactive screen forms and such.

SC24 (Graphics):

No work – even though SC24 looks like the obvious place to put windowing standardization.

It is an article of faith that no international standard may encroach on another's

territory, and that the terms of reference of each SC do not overlap. This presents difficulties in dealing with new (well, new in ISO terms) and widely-applicable technologies such as windowing. Perhaps it may be possible to hand the issue to SC24 without upsetting other SCs. Alternatively, it may be necessary for JCT1 to set up a whole new SC to run with it, and bring the currently fragmented work together. (This recently happened on security issues – see below.) Again, watch this space for more news.

9945-2 Shell and Tools Standard

The majestic machinery of JTC1/SC22 has sanctioned the use of draft 9 of IEEE 1003.2 as a draft proposal (DP), which embarks forthwith on a six-month balloting period. This period is to be synchronized with the IEEE's ballot, with the result that 1003.2 and 9945-2 move forward in lock-step, and should hit the streets simultaneously as identical American and international standards.

Document Format

In order to avoid future wrangles over document format with ISO's Central Secretariat, and to avoid time wasted in recasting IEEE standards into ISO's mold, *all* 1003 standards are to be created and balloted in a format acceptable to ISO. (And to the IEEE. And to the POSIX working groups. But mostly to ISO.)

WG15 is concerned that ISO's standards for standards were drawn up with relatively short documents in mind. For example, ISO's Central Secretariat objects to the line numbers which appear in draft 13 of 1003.1 – even though it used the line numbers in referencing other changes that it wanted! Hopefully, an acceptable compromise will be reached. Working group chairs and editors will be told what the changes mean to them just as soon as a decision is reached.

Rapporteur Groups

The concept of rapporteur groups is an ISO invention. It refers to a group of "technical experts" (another ISO term) from a number of related standards efforts, or concerned with

ISO	IEEE	Topic	Notes
9945-1	1003.1	OS interface	Now
	1003.1a	Clean-up	1990
	1003.1b	Extensions, language independence etc.	Future
	1003.4	Real-time	Future
	1003.6	Security	Future
	1003.8	Transparent file access	Future
9945-2	1003.2	Shell & tools	First release
	1003.2a	User Portability Extension	Future
9945-3	1003.7	System administration	First release
1xxxx-1	-	C bindings	Future (probably to be done by new 1003 working group)
1xxxx-2	1003.5	Ada bindings	Future
1xxxx-3	1003.9	FORTRAN bindings	Future
-	1003.0	POSIX environment	Some overlap with ISO DP 10000, <i>International Standardized Profiles</i>
-	1003.3	Test methods	Under consideration by rapporteur group
-	1003.8	(Aspects besides T.F.A.)	Work elsewhere in ISO on RPC
-	1003.10	Supercomputing	Profile: relevant to DP 10000
-	1003.11	Transaction Processing	Profile; also relevant to SC21/WG3 database work
-	1201.x	X Window	See below
-	1224.x	Interfaces to OSI services	Not clear where these fit in ISO work: SC21 (OSI) seems to be against working on bindings

Table 1: Correspondence between ISO and IEEE Activities

a specialized topic within a single standards effort, which meets to discuss its area of interest. Members of the group then report back to their own groups, in order to integrate the work of the rapporteur group and the standards efforts that it links.

WG15 has three rapporteur groups: Conformance, Internationalization, and Security. Each addresses areas known to have applicability in fields broader than POSIX itself. For example, JCT1 has just created a whole new subgroup (SC27) to handle security, bringing together separate developments in SC18 (Office systems), SC20 (Data encryption), SC21 (Open Systems Interconnection), SC22 (Languages)* –

and anything else which turns out to have security implications. (I mentioned this development in my last report, but managed to garble some of the references. Sorry about that...) Similarly, there is work on conformance testing and internationalization both inside and outside ISO.

In Brussels, the rapporteur groups all held informal meetings separate from the main business of WG15. Since all three have only just gotten off the ground, there is little to report as yet, but watch this space!

X/Open Portability Guide as a European Standard?

At the May meeting of WG15, our minds were much exercised by a proposal from CEN (Comité Européen pour la Normalization –

* Why is the POSIX project a subdivision of the languages subgroup? Because it was the least unsuitable place in the ISO structure to put it at the time...

The European Committee for Standardization) that the whole of the third edition of the X/Open Portability Guide (XPG3) should become a draft European prestandard. The arguments against doing this center on the fact that the XPG is not a formal standard reached (slowly) through consensus, but an informal document which references formal standards where it can, but which then goes on to fill the gaps with *de facto* and suggested standards material. Increasingly, the European countries which form CEN's membership have come to realize that a document of this type, while useful in its own right (arguably more useful than existing formal standards, in fact), cannot be adopted as a European standard for both legal and practical reasons.

XPG3 has, however, helped to focus European minds on areas where formal standards are lacking. At the moment it looks as though the CEN project charged with producing a POSIX standard will build on the output of WG15. In addition to this, Germany is in favor of adopting as prestandards those parts of XPG3 which do *not* correspond to existing or emerging international standards – for example, ISAM, curses and X Window. The argument for this is that some kind of standard is urgently needed in these areas. The argument against, coming from Britain, Denmark, the Netherlands and others, is that CEN can only adopt standards which are public – *de facto* just isn't good enough, and besides, such things are outside the scope of the original work order for a POSIX standard. At the moment, it looks as if this point of view will prevail.

As a sidelight to this issue, it seems that ISAM will eventually make it into the POSIX standard, as X/Open has expressed a desire to submit a base document to the 1003.1 working group.

Harmonization and Synchronization

The three previous headings – 9945-2, rapporteur groups, and the CEN standard for POSIX – highlight a couple of important issues identified by JCT1:

Harmonization:

Standards covering identical or related topics should be in agreement; and

Synchronization:

Development work on standards covering identical or related topics should be developed in step with one another, both so that there is no unnecessary delay between the appearance of one standard and the appearance of another, and to avoid duplication of work – for example, the same ballot objection being made to and fielded by two separate groups.

WG15 has taken steps to synchronize its activities with those of the IEEE 1003 working groups, its main feeder. In some cases this means that WG15 will set IEEE timetables – almost a case of the tail wagging the dog, but necessary in order to arrive at international standards as quickly as possible.

To address the issue of harmonization, WG15 discussed a new category of liaison to JCT1. Liaison is a mechanism which allows transnational and international setters and users of standards to monitor or to contribute to the work of ISO. Participation is otherwise the province of national standards bodies such as ANSI, JISC and DIN – ISO is currently bad at dealing with regional standards bodies such as CEN. The proposal embodied a combination of sticks and carrots which would allow other types of standards bodies to participate on the condition that they undertook to align with relevant international standards within some reasonable time after publication. The working group reached no conclusion on this radical idea, and will discuss it again at its next meeting.

It will be a while before JCT1 gets around to considering any proposal of this nature. In the meantime, WG15 will continue to invite observers such as myself to its meetings.

Language Independence

As at the previous meeting, this topic was discussed at some length. The policy of JCT1 is that, ultimately, in the interests of precision and verifiability, all base standards should be written in some formal language which is itself the subject of an ISO standard. There is a small problem here: no formal language

suitable for use in the POSIX project is yet the subject of a standardization effort (although IEEE P1003.7, System Administration, is making use of ASN-1, a standardized formal language developed for use in describing communications systems). If POSIX were to wait for a formal language to be standardized before breaking the current links between POSIX and C, nothing could be done for a couple of years. However, it is necessary to break the links with C as soon as possible, in order that additional bindings for Ada and FORTRAN can be defined. The break will be made informally, by using English along with language-independent data types, and so on.

In parallel with this development by WG15, a research project funded by the European Community (EC) looks like it will be funding the development of a description of the POSIX operating system interface in VDM-SL (Vienna Definition Method Specification Language). SC22 is actually thinking about standardizing this formal language, which is already being used in the production of an ISO standard for Modula 2 by SC22/WG13. Welcoming what is, in effect, an offer to discover the problems involved in defining POSIX using a formal language, WG15 has sent a message of encouragement to the EC, while emphasizing to SC22 that, as far as POSIX is concerned, the coming language-independent description is a necessary step on the path towards a formal definition.

The Portable Common Tools Environment

Another research project supported by the EC concerns the Portable Common Tools Environment, PCTE. Essentially a very sophisticated and all-encompassing object-based workbench for the support of

Computer-Assisted Software Engineering (CASE), PCTE is the result of six years' work, and the investment of several million European Currency Units (ECUs) by government and industry - with more years and mega-ECUs to come. Among other organizations, NATO is a strong champion of the technology. The European Confederation of Computer Manufacturers (ECMA) has, over the last couple of years, been working on a PCTE standard which may (just) be ready in 1991, and which may then be offered to ISO.

What has this to do with POSIX? Well, PCTE was originally aggressively host-independent - independent, that is, both of hardware and, on systems where it was not to run native, of operating system. This made excellent sense six years ago when development started - using UNIX as a development host. Versions are currently available for several UNIX hosts, with VMS and IBM mainframe versions on the way. Times move on, however, and there is now (ISO Central Secretariat permitting) an international standard hardware-independent operating system which looks like it will become the predominant host for PCTE. It makes sense, therefore, for PCTE to align itself closely with POSIX, so avoiding unnecessary duplication or conflict of functionality. Following a morning of presentations by PCTE experts, WG15 agreed to keep members of the ECMA PCTE working group informed of its activities.

Next Meeting

The next meeting of WG15 is to be held in Paris, France from June 11-15, 1990, and is to be hosted by AFNOR, the French national standards body.

An Update on UNIX and C Standards Activity

Jeffrey S. Haemer

Report Editor, USENIX Standards Watchdog Committee

IEEE 1003.0: POSIX Guide Update

An anonymous correspondent reports of the April, 1989 meeting:

The April session of 1003.0 was fruitful. The most significant accomplishment was the proposal and development of definitions the committee feels it needs to describe an open systems environment properly and adequately. Five definitions were developed:

- open system environment
- application environment
- application environment description
- application environment profile
- POSIX open system environment

Group consensus was that the first four would be submitted to the JTC1 Application Portability Study Group as a draft proposal for its work. The committee added the caveat that these were draft definitions, subject to change. A key clarification by these definitions was the distinction between an application profile and an open system environment: a profile is a subset of the environment.

The guide document, being developed by 1003.0, is nearly mature. Significant strides were made in the architecture section, which focuses on the operating system interface, languages, and network services. In the following months, 1003.0 will turn its attention to database management, data interchange, and graphics. The user interface section will be closely coupled to the work of the newly formed, IEEE 1201.1 (Xwindows) working group. Similarly, the transaction processing section will track the on-line transaction processing (OLTP) group (1003.11).

There is some worry about the length of the guide – currently 135 pages and growing. If the document becomes unwieldy, some attention will be turned to scaling it down.

The committee also created an Internationalization study group, to cut across groups and help increase inter-group coordination in this area. The study group intends to become a full working group in Brussels, this October.

IEEE 1003.0: POSIX Guide Update

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the July 10-14, 1989 meeting in San Jose, California:

As 1003.0 passes the mid-point of calendar year 1989, progress can be earmarked by the arrival of line numbers to the guide document. I remember the first time I saw line numbers on a document within the IEEE 1003 arena. My first thought was "this committee is really doing precise, exacting work." Thus was my reaction again when I saw line numbers on our document. My balloon was burst, when one of our active members – and by "active member" I mean someone who commits contributions in writing, not just someone who comes to voice an opinion in a talk-show-like atmosphere – pointed to our ISSUE LOG, which states that the committee needs to do more work. (There's that word again.) Alas, I came back down to earth. I have "miles to go before I sleep."

Dot Zero continues to converge. Our document is finally beginning to tie together the standards and elements that comprise a POSIX open system. Key events continue to be the definition of terms that will eventually make it to the IEEE Glossary and the identification of areas where terms still need definition.

The group is still generating discussion/debate/argument/food-fights over behemoth macro-questions such as, "What is the role of the guide?" and, "What is the PROPER audience?" In addition, the group has made valiant attempts at addressing specific areas such as graphics and data interchange without the benefit of focused expertise. We now agree on

our ignorance in these areas, and will seek help and/or point to other committees that, we believe, can come up with the answers.

Overall, we must meet our objective of going to ballot in October 1990, because that is what I told my wife, who is still trying to figure out what in the world a "dot zero" might be.

IEEE 1003.1: System Services Interface Update

Shane McCarron <ahby@bungia.mn.org> reports of the April, 1989 meeting:

"After thinking about it, I realized that 1003.1 did actually do some stuff this quarter." [April -ed]

1003.1 is preparing two supplements, A and B, to 1003.1-88.

At the 1003.1 meeting in Minneapolis, the group reviewed draft 0.1 of 1003.1, supplement A. This supplement contains only clarifications and editorial comments, and will be balloted in the Summer. It will be provided to the ISO as the United States' comments on the International Standard IS 9945, which is the same as 1003.1-1988. Its goal is to ensure that the ISO standard and the IEEE standard (with supplement) are functionally identical.

Supplement B, to be balloted later, contains substantive changes: new facilities absent in IEEE Std 1003.1-1988. Some were missing from 1003.1-88 because they weren't completely specified in time to be included in the first release of the standard. Others are being introduced due to requests from other standards committees or the user community. For example, the ISO working group responsible for POSIX has requested a new archive format. It argues both that the archive formats in the first standard are insufficient for the future needs of POSIX systems and that a dual solution is unacceptable. The new format uses ANSI standard labeling, but extends it to permit POSIX filenames, security information, etc.... Supplement B also includes symbolic links, `truncate()`, `ftruncate()`, `putenv()`, `clearenv()`, `getpass()`, `seekdir()`, `telldir()`, `chroot()`, `chmod()`, `chown()`, and `fsync()`.

Supplement B will also contain additional clarifications and edits to the base standard. The ISO will probably designate this supplement an addendum to IS 9945. All this maneuvering ensures that the different standards stay in sync, and prevents large delays in getting the ISO standard approved.

Although 1003.1-88 is now official, the 1003.1 committee's work will continue for some time yet. As other POSIX standards gel, their committees uncover requirements for additional functionality or semantics in the base standard, to support their work. As these committees point out such cavities in the standard, P1003.1 works to fill them. Everyone's hope is that no root canals will be necessary.

IEEE 1003.3: Test Methods Update

Doris Lebovits <lebovits@attunix.att.com> reports on the July 10-14, 1989 meeting in San Jose, California:

Overview

This was the thirteenth meeting of P1003.3. Monday through Wednesday, the group began work on a verification standard corresponding to 1003.2 (Shell and Tools). Following the close of the formal meeting, the technical reviewers of the draft 10 ballot met for the remainder of the week.

Meeting Summary

This was the first meeting to develop the verification standard for P1003.2, which will contain test methods and test assertions for measuring 1003.2 conformance. This standard will ultimately form part III of P1003.3. (Part I contains definitions, generic test methods, and so forth; part II is test methods for measuring P1003.1 conformance, including test assertions. As other P1003 standards reach maturity, their verification will, in turn, be covered in new parts of the P1003.3 standard.)

The chair's aggressive goal is to be ready to bring part III to ballot after four quarterly meetings. A detailed schedule and milestones will be developed at the next meeting.

Attendees included representatives of AT&T, NIST, OSF, Mindcraft, IBM, DEC, HP, Data General, Cray Research, Unisys,

Perennial, and Unisoft Ltd. The meeting agenda included:

- the confirmation of new officers for the part III work

Chair: Roger Martin

Vice-chair: Ray Wilkes

Technical Editor: Andrew Twigger

Secretary: Lowell Johnson

- the rough scheduling and setting of general milestones for part III
- a meeting with the P1003.2 working group to discuss testing issues
- action item assignments
- identification of items for the next meeting

In addition, small groups formed to discuss and resolve three specific issues. One group investigated the difficulty of thorough testing of the more complex utilities: *awk*, *bc*, *ed*, *lex*, *make*, *pax*, *sh*, and *yacc*. (The resulting action item was to produce a prototype set of assertions.) A second group scoped the writing of assertions for BNF type structures: [] expressions, regular expressions, and extended regular expressions. The third reviewed "Verification of Commands Interface," a paper by Andrew Twigger of Unisoft Ltd. The paper covers:

- character set and locale
- internationalized utilities
- underlying OS primitives
- regular expression testing
- pattern matching notation
- utility syntax rules
- errors from P1003.1 associated functions
- environment variables
- standard output format
- standard error format
- environmental changes
- symbolic limits
- obsolescent features
- job control
- read-only variables
- signal numbers

NIST has contributed its current 1003.2 test assertions to provide a basis for the 1003.2 verification work. Sheila Frankel of NIST gave a short presentation on the current state of

these assertions, which include approximately 900 Mindcraft test assertions, plus 2600 newly-created assertions, all based on P1003.2 Draft 8.

Technical Reviewer's Meeting

In parallel to the verification work for P1003.2, balloting and revision is taking place on draft 10 of parts I and II.

As of July 6, 1989, 77 responses had been received from the 125 members in the balloting group. Eighteen additional responses will bring this to the 75% response needed to officially close the ballot.

The tally of the 77 responses:

28 positive	(36%)
31 negative	(40%)
18 abstain	(24%)

The technical reviewers held a plenary session to evaluate and respond to the comments and objections to this draft. Group consensus decided each issue and each decision was final. Part I was reviewed completely but only a few chapters of part II were reviewed. The remaining part II work was assigned to volunteers.

Draft 11 is planned for ballot recirculations in October, 1989, and an approved standard for parts I and II is anticipated by the first quarter of 1990.

IEEE 1003.4: Real-time Extensions Update

John Gertwagen <jag@laidbak> reports on the July 10-14, 1989 meeting in San Jose, California:

The P1003.4 meeting in San Jose was very busy. The meeting focused on resolving mock ballot objections and comments. Despite limited resources for documenting changes, a lot of work got done. Here's what stood out.

Shared memory

The preferred interface falls somewhere between shared-memory-only and a mapped-files interface, such as AIX's *mmap()*, which allows files to be treated like in-core arrays. Group direction was to reduce the functionality to support only shared memory,

so long as the resulting interfaces could be implemented as a library over *mmap()*.

Process memory locking

The various region locks were clarified and, thus, simplified; the old definitions were fuzzy and non-portable. For those who haven't seen it, there is actually a memory residency interface (i.e., fetch and store operations to meet some metric) instead of a locking interface. Most vendors will probably implement it as a lock, but some may want it to impose highest memory priority in the paging system.

Inter-process communication

Members questioned whether the interface definitions could really support a broader range of requirements; they're like no others in the world today. Having been designed to meet the real-time group's wish list, there are lots of bells and whistles – far more than in System V IPC – but it's not clear, for example, that they are network extensible. Discussions in these areas continue.

Events and semaphores

Members were concerned about possible overlap with other mechanisms, especially those being considered for threads. The question is basically, "Should there be separate functions for different flavors or a single function with multiple options?" General sentiment (including our snitch's) seems to be for multiple functions; however an implementation might choose to make them library interfaces to a common, more general system call. There is, however, a significant minority opinion the other way.

Scheduling

Many balloters found process lists and related semantics confusing. An attempt was made to re-cast the wording to be more strictly in terms of process behavior.

Timers

Inheritance was brought in line with existing (BSD) practice.

* * *

Outside of the mock ballot, there were two other major news items.

First, there is a movement afoot to make the .4 interfaces part of 1003.1. They would become additional chapters and might be voted separately or in logical groups. This would bring P1003 in line with the ISO model of a base standard plus application profiles. P1003.4 would become the real-time profile group. This is a non-trivial change.

Up to now, the criterion for .4 has been that of "minimum necessary for real-time," and has coincidentally been extended to support other requirements "where convenient." This is not a good starting point for a base interface. For example, *mmap()*, or something very much like it, is probably the right base for "shared storage objects," but real-time users want an interface for shared memory, not for mapped files. Our snitch worries that things might look a bit different had the group been working on a base standard from the beginning.

Second, the committee officially began work on a threads interface, forming a threads small group and creating a stub chapter in the .4 draft. A working proposal for the interface, representing the consensus direction of the working group, will be an appendix to the next draft.

A lot of work remains to be done before .4 can go to ballot and the current January '90 target may not be realistic. If the proposed reorganization occurs, a ballot before the summer of 1990 seems unlikely.

IEEE 1003.5: Ada-language Binding Update

Ted Baker <tbaker@ajpo.sei.cmu.edu> reports on the July 10-14, 1989 meeting in San Jose, California:

The Ada-language binding for 1003.1 is progressing steadily, though behind schedule. The group agreed to try to prepare a document for the October meeting in Brussels that is ready for mock ballot. Those at the meeting will decide whether the document has achieved this goal. If not, we will try again at the January meeting in New Orleans.

The slow progress is mainly due to the long time between meetings and the limited work force available to do the writing. The members, all volunteers, must steal time for POSIX from their "real" (i.e. paying) jobs. Attending quarterly meetings already puts most members near the limit of time they can spare.

Most significant technical issues seem to be resolved; the remaining controversies center on almost-religious issues, such as the exact grouping of interface declarations into Ada packages, naming, capitalization conventions, and where to strike the balance between providing full functionality and idiot-proofing the interface.

Each chapter has been assigned to a person for review and editing, based on decisions made at the San Jose meeting. Quite a lot of writing still needs to be done. Chapter 7 ("Device- and Class-Specific Functions" – i.e. terminal interfaces) is still empty, and some others are still mostly just Ada code, with no discussion. Most of the rationale remains to be written. Mitch Gart has agreed to coordinate this, including a chapter on "meta-issues" – design decisions affecting the entire interface. David Emery will combine the chapters to produce the next draft.

Interaction with 1003.4 (Real-Time Extensions) has heated up, with 1003.4's consideration of support for multi-threaded processes. Ada language implementations must support multiple tasks (i.e. threads) within a POSIX process, to comply with the Ada language standard. Neither the 1003.1 standard nor the 1003.4 draft that just completed mock balloting supports multithreaded processes, so the Ada implementor is currently forced to hack out some sort of internal concurrency scheme, with its own layer of dispatching, for each Ada process. This tends to run aground when one Ada task makes a blocking system call, since the whole process is forced to wait. Naturally, Ada implementors and users would be pleased if the POSIX interface provided for concurrency within a process.

The Ada group is very interested in the threads proposal, and most members would like to see some support for threads in the 1003.4 standard that goes to formal ballot.

Some members are a little bit concerned that those working on the proposal may not understand Ada tasking well enough to ensure that the proposed threads will be adequate to implement Ada tasking semantics. This has been very frustrating for members of the Ada group, since the discussions of the threads proposal were all in parallel with meetings of 1003.5. The best the Ada group was able to do was to keep one observer present (on rotation) at the review of the threads proposal. It is not clear whether this was adequate.

[Editor's note: What's going on here, and in the second paragraph, is that some groups are much larger than others. 1003.5 is among the smallest. The 1003.4 session I saw had about forty overworked attendees. The 1003.5 sessions I saw had five to ten.

1003.5 could use a lot more participation from the Ada community. Unfortunately, this may be a case of "once burned, twice shy." For years, there's been a lot of talk about "Ada environments," all of which seem, from a UNIX perspective, like enormous, cumbersome projects that might actually come into widespread use in, if not our children's lifetimes, perhaps their children's.

Make no mistake about it: the Ada community is huge. And easy availability of machines with implemented, Ada-language bindings to POSIX-conformant operating systems would be immensely useful to that community. The ability to buy a box, off-the-shelf, with a portable environment for running Ada programs in the next couple of years, would make Ada programmers' lives immensely easier and even be a big aid in implementing the richer and more complex environments mentioned in the previous paragraph.

Still, you can guess what the average, UNIX-naive, Ada programmer must think: "Whoopie, another standard/environment. I'll have to take a look at it in a few years to see how it's coming along." If the IEEE could make some non-vanishing fraction of the Ada community understand that POSIX is on the verge of being here, now, dot 5 might get a lot more help.

This seems to us (that's the editorial "we," folks) like a quintessential marketing problem. If 1003.5 could enlist the help of 1003.0 in this matter, they might be able to make some real headway here.]

The 1003.5 group is also very interested in the progress of the language-independent versions of the POSIX standard. Much of the labor of the Ada binding group has been

devoted to separating the essential semantics of the 1003.1 interface from the details of its expression in the C language (for example, *setjmp()*, *longjmp()*, and signal handlers). This labor may be of use to those working on the language-independent version of 1003.1, but the Ada group does wish that new standards, such as 1003.4, would start out with a language-independent document, rather than adding to the language-bias problem.

There was one change in the leadership of the 1003.5 working group. Stowe Boyd, of Meridian, had been vice-chair but is no longer able to spare time from his job to work on this project. Steve Deller, of Verdix, has agreed to replace him. This is a very important job, since the vice-chair of the 1003.5 group takes direct responsibility for setting the technical agenda and running meetings.

IEEE 1003.6: Security Extensions Update

Ana Mariá de Alvaré (anamaria@llc.llnl.gov) reports of the April, 1989 meeting:

P1003.6 covered these global issues at the five-day Minneapolis meeting:

1. Supplements to 1003.1 will address portability, data interchange format, and symbolic links. This means 1003.6 must also consider those areas.

2. 1003.6 would like to define a system variable that tells what security policies are allowed on the system, and a function that returns which security-related attributes (e.g., MAC, ACL) are currently in operation. Such changes would need to be made in collaboration with 1003.1.

3. Other pieces of 1003.1 and its supplements may conflict with security extensions. A short-term subgroup was proposed to review these documents and propose additions or changes. 1003.6 is looking for volunteers for this work.

[Ed. – If you have, or can imagine, the orange book and the ugly green book side-by-side on your bookshelf, now's the time you should work to ensure that only their colors clash. The chair of 1003.6 is Dennis Steinauer, who, we believe, would be happy to hear from you if you're willing to help (steinauer@ecf.ncsl.nist.gov).]

4. Two members of the networking group (1003.8) joined 1003.6 for half a day to list and explain their areas of concern: transparent file access, authentication at mount time, setuid programs, file format, local id contents, and who does the audit. These issues were scheduled to be revisited at the San Jose meeting in July in a joint meeting of the two groups.

5. Charlie Testa gave a status report on TRUSIX. The TRUSIX working group responded to Tom Parenty's paper, which summarized the TRUSIX efforts. The members felt compelled to clarify certain sections that they believed misconstrued their real objective: the creation of a trusted UNIX operating system. This response is also published in the December, 1988, Data Security Letter Journal.

There are serious conflicts and points of contention between POSIX and TRUSIX. POSIX is worried that systems conforming to TRUSIX recommendations will get preferential treatment during product evaluation, that vendors who currently plan only Class B2 systems or below are excluded from TRUSIX, and that participants in TRUSIX share proprietary information. TRUSIX takes the position that the marketplace should be the final judge. TRUSIX will be POSIX compliant, and will make no attempt to force vendors to be both POSIX and TRUSIX compliant. If customers force a *de facto* standard of dual compliance for even non-DOD applications, so be it.

TRUSIX's ACL proposal will be delivered to the IEEE at the July meeting. The proposal is only a guide, and it will not be written in a formal specification language as a favor to the reader.

TRUSIX's audit subgroup is trying to follow both POSIX and X/Open efforts in this area. Their subgroup is focusing on pre-selection, in contrast to the 1003.6 focus on post-selection, and they will review a token-based scheme at their next meeting.

6. At the previous meeting, a common descriptive top-level specification language (DTLS) was proposed. For the moment, this language will form an appendix to the draft, and will be used as an internal tool to let the

group define unambiguous security interfaces. Every subgroup of 1003.6 will provide descriptions of interfaces in both English and DTLS. Steve Sutton will be the chair of the DTLS team, and will work in conjunction with the technical editor of the draft.

The Security Working group is split into separate groups for audit, discretionary access control (DAC), mandatory access control (MAC), and privileges. Each subgroup gave a summary report at the end of the week and some were able to give a first-cut delivery schedule. The following is a short summary of each group's efforts.

Audit

The scope of the audit group encompasses audit definition, auditable events, audit trail contents, and audit trail access and control. The group will also define a portable audit trail data representation and focus on post-processing event classes.

Audit records will include process identification, audit id, effective user id, effective group id, media addresses, MAC labels, and privilege information. In San Jose, the audit group will try to identify all token types, define the audit id, propose some changes to the "seek" function, pursue event classes, and review and merge the DTLS interface descriptions with the English sections.

DAC

The DAC group is almost done with its rationale section. One question this time around was how to pass access mechanisms based on DAC across the network. Currently, file ownership is the first access check; on networked systems, this can lead to spoofing, particularly when root tries to access files on other systems.

Another hot issue was access functions. The consensus is that an access function to an opaque DAC (i.e., one that prevents knowledge of the structure) should replace the use of *stat()*, *chmod()*, *stat()* or locking mechanisms for controlled file access. The function will not replace *chmod()*, *stat()* or permission bits; however it will define operations that will allow applications to continue to work correctly in the face of ACLs.

MAC

Issues addressed here come from the MAC requirement that all system objects be labeled with security levels (e.g., CONFIDENTIAL, SECRET, TOP-SECRET). Two proposals were on the table – one from Addamax, the other from Olin Sibert – but no strong consensus was reached. Miscellaneous comments on the issues discussed:

1. Downgrading (of security levels)
 - How should it be done?
 - Must the old label dominate the new?
 - Does downgrading need to be strictly controlled?
 - What about upgrading?

2. Directory labels.

mkdir should be allowed to label directories on creation, to permit portable, level-hierarchy-dependent applications.

3. File locking.

The standard should address locks and may consider them as objects.

4. "Write-up" appends.

Writing to a file at a level above you is known as "write-up." Processes can write to files that they can't read. At first blush, this seems analogous to standard UNIX, which allows files with permissions `-w-w--w-`. What MAC adds is the prohibition that the process even know if the write succeeds. Because appending to such a file provides no way to assure that the write succeeded, the question of whether to allow such write-ups was raised and discussed.

5. Change of file level with open file connections.

UNIX does not expect open connections to break. (An exception is */dev/tty** on 4.3 BSD, which can be checked for open connection breaks.) Since */dev/tty** are special files and 1003.1 doesn't address special files it was argued that 1003.6 need not either, but this issue will be discussed further in San Jose.

6. Open tranquillity.

The tranquillity property states that a resource should not be in active use during changes to its attributes. (See also issue #5 above.) It was stressed that POSIX should be defining states and mechanisms that are as safe as

possible, obvious to implement, deterministic, and clear. Only privileged processes should be able to change the MAC label of a file object.

7. Replication or Recalculation?

Replication means copying current properties across from one label to another. Recalculation means re-evaluating the situation, then assigning properties or attributes needed for each file to work as labeled. The consensus was that recalculation is needed in the standard, but there was no consensus on how either recalculation or replication should occur.

8. Multilevel directories

A "multilevel directory" is a directory with files at different levels (e.g., both TOP SECRET and CONFIDENTIAL). Should a multilevel directory feature be available for general use? Should it be part of the standards? If so, operations on multilevel directories would be restricted and functions to be able to create, check for existence, and query for directory name would be required. These directories would inherit their DAC from their parent.

The directory that stores files the user can see at the current time, as determined by the label at request time, is the "access hidden directory." An open question is whether access to such a directory should be controlled by process privilege or the pathname syntax.

9. Text Format

Two proposals were put forward on text format, but only one was discussed because of time constraints. Despite this, the group resolved that naming should be site-specific, but names should be unique and order-independent. Furthermore, a label should be interpretable and unique. One major problem was that the characters suggested for hidden directories were outside the constrained character set provided by 1003.1 – [a-z][A-Z][0-9] and a very limited set of punctuation characters.

10. System High/Low?

This government concept is used a lot in discussions of secure systems. It was put on the agenda for the July, San Jose meeting.

11. Other Issues

Should the standard assure a non-decreasing directory hierarchy? In other words, should subdirectories always have at least as high a level as the parent? Should the standard define level ranges such as system high? Should the standard define a process clearance range? (Clearance only defines how to specify an error return that the system is allowed to give.)

Privileges

The group reviewed interface functions defined at the previous meeting, and agreed on all of them except *exec()*, which poses unresolved problems about inheritance of privileges. The group expects to finish this in July.

Some of the functions defined so far are:

- is_effective(p)
- make_effective(p)
- make_ineffective(p)
- is_inheritable(p)
- make_inheritable(p)
- make_not_inheritable(p)
- is_permitted(p)
- relinquish(p)
- make_effective_if_inherited(p)
- make_all_ineffective(p)

all related to querying the process privilege state.

Old goals were revised and new goals added, including: support for old binaries, support for new binaries implementing true least privileges, acquisition of effective privilege following *exec()*, prevention of some programs from inheriting privileges, and unsetting of privileges on exit from signal handlers.

Other issues included:

1. Privilege inheritance

When is it needed?

2. Forbidden privilege

Should a flag be available to forbid a process to gain a privilege?

3. Privilege System Variable

Should the standard define a system variable to set privileges at installation time?

IEEE 1003.6: Security Extensions Update

Ana María de Alvaré <anamaria@llc.llnl.gov> reports on the July 10-14, 1989 meeting, in San Jose, California:

P1003.6 (security) is split into four main groups: privileges, mandatory access control (MAC), audit, and discretionary access control (DAC). In addition, there is a definitions group, whose charter is to define terms and to ensure that definitions used by 1003.6 do not clash with definitions in other 1003 groups.

Definitions

The definitions group reviewed all definitions new to draft two. The majority were from the audit group and were approved. Amusingly, the lone exception was the definition of "audit," which included an interpretation of an audit record; the definition group considered this to be outside the audit group's goals.

The group also chose a global naming convention, *PREFIX_FUNCTIONNAME*, where *PREFIX* represents the security section/topic. Current prefixes are "priv_," "mac_," "aud_," and "acl_" (DAC). The same prefix rule extends to data structures (e.g. "priv_t").

MAC

Several issues were resolved.

- A "write up" standard will be neither restricted nor guaranteed.
- The "upgrade directories" function was dropped, since a "write up" without a read does not guarantee success.
- Change file label/level and change process label operations will be accepted for privileged processes.
- The MAC_PRESENT variable will be added to the sysconf, to indicate that a MAC mechanism is installed in the system. MAC_CONTROLLED and MAC_ALWAYS were also proposed. MAC_CONTROLLED would return the value of a file controlled by a MAC mechanism, and MAC_ALWAYS would indicate that all objects on the system contain associated MAC information.

- A set of six privileges were defined:

P_upgrade
P_coverchannel
P_MAC_READ
P_MAC_WRITE
P_LABEL_OBJ
P_LABEL_SUBJ

The last two might be folded under READ/WRITE privileges, however these two are the most sensitive of all.

The next meeting will see discussions of Sun's multiple-level directories, the recalculation function, and information labels. The group will also review the .6 draft, the MAC common description language interface, and 1003.1/1a.

Privileges

The privilege group has defined interfaces for file privileges. For example, *priv_fstate_t()* will return whether privilege for the file is required, allowed, or forbidden. A process's privilege can be permitted, effective, or inheritable.

Also, there is now a list of needed privileges, including PRIV_SETUID and PRIV_SETGID (set the uid and gid of a file or process), PRIV_FOWNER (change the owner uid of a file), PRIV_ADMIN (do administrative operations like unlinking a file), PRIV_RESOURCE (set the sticky bit or be able to use memory), and DAC_READ/WRITE (override access search or read and access write).

The process-privilege interface is still an open issue, and will be discussed in October. These three suggestions are on the table:

1. A function pair. *priv_set_priv(id, attr, value)* and *priv_get_priv(id, attr)*. (Something of type "value" can take on the values "required," "allowed," or "forbidden.")
2. An interface to set or unset multiple privileges at a time.
3. A requirement that the operating system recalculate privileges for each process every time that process manipulates an object.

Next meeting, the privilege group will focus on developing functional interface

descriptions in both English and in Common Descriptive Language (CDL).

DAC

The DAC group decided to describe interfaces using a procedural interface. They defined the minimum set of functions required for access control lists (ACLs) – open, close, write, sort, *create_entry*, *get_entry*, *dup_entry*, *delete_entry*, *set_key*, *get_key*, and add/delete permission – and the minimum set of commands – *getacl* and *setacl*. They also defined the needed privileges and passed their list to the privilege group. The October meeting will focus on polishing the current draft and addressing default ACL interfaces.

Audit

The group discussed portability, especially data portability. Should only privileged processes write to audit logs? (The consensus is, "Yes.") And how much should the record format be standardized?

The October meeting will see a draft review, plus discussions on event identification, classes, style and data representation, and token grammar.

New Group: Network/System Administration

Because interconnectivity is at the heart of many security and administration issues, "interconnectivity" between P1003.6, P1003.7 (system administration), and P1003.8 (networking) had to improve. A joint evening meeting of the three groups set this in motion, and five members of 1003.6 have signed up to review drafts from the other two groups. They intend to begin working on this area formally in October.

IEEE 1003.7: System Administration Update

Steven J. McDowall <sjm@mca.mn.org> reports on the July 10-14, 1989 meeting, in San Jose, California:

War and Remembrance – How I survived a POSIX Meeting

Listen closely to this tale of wonder and bewilderment and hope that you shall never have to face such horrors as I. Yes, I was

there when, in a flurry of activity, the 1003.7 committee elected Steven Carter to the chair. To show he was a good choice, Carter immediately sat on the chair to which he'd been elected. This was swiftly followed by the election of Vice-chairs Martin Kirk and Dave Hinmant (though I shall speculate not on what vices they may have perpetrated on those chairs); Mark Colburn, Secretary (owing to a proven ability to take dictation lying on a pool-side sun bed); and their honors Bob Bauman and Shoshana O'Brien, Technical Editors.

You may sense that I feel few exciting things happened in San Jose. Correct. I wish this group would get into some real fights, like other groups. Interoperability may prove our only hope. Still, progress is progress, however uncontentious. Here's what else seemed to me to be important.

1. Language Independence

The group voted, nearly unanimously, that the country of Language should be independent. We were uncertain about where, precisely, it might be, but tentatively put it near Borneo.

We chose to use ASN.1 ("Abstract Syntax Notation – 1") as our internal notation for data structures. The group also appointed me representative to the 1003.1 language-bindings group to watch what those pursuers of knowledge are doing in this area.

2. Interoperability

X/Open continues to push this into the foreground. Luckily for us, they also continue to help us understand what it entails. Group consensus holds that interoperability is within the purview of 1003.7. What we're still uncertain of is how far down we should standardize; only through the application layer? down to the packet layer?

For example, a standard application-layer protocol ensuring interoperability might require that certain Application Program Interface (API) calls be available, with given arguments and results, but say nothing about how those calls are made. In contrast, a transport-level protocol might require that the information be fed into the API will be in a pseudo-ASN.1 format to help in non-homogeneous networks. A still lower level protocol might detail the exact

packet structure, including ASN.1 format for the object data, to prevent foreign machines in a non-homogeneous network from throwing out otherwise unrecognizable packets.

Most committee members have strong, idiosyncratic ideas about this subject and the issue is certain to resurface in Brussels. We need input on this from the community at large. Where do YOU think a standards organization like the IEEE should draw the line in ensuring interoperability?

[Editor's note - This is not a rhetorical question. Things you do in the future may be affected by decisions P1003.7 makes in this arena. If you have an opinion on this subject, speak up.]

As an aside, the current X/Open representative, Jim Oldroyd of the Instruction Set, Ltd., who has really helped the group a great deal in this area, may not attend the next 1003.7 meeting. We think this would be a real loss, and hope that X/Open and his employer find a way to arrange for him to go.

3. Misc.

Some progress was made in doing the ASN.1 syntax for a few of the basic objects the committee decided on for phase I of the standard. Everyone is discovering that defining such objects (File Systems, Devices, Spools, etc.) in a non-ambiguous way using a meta-language like ASN.1 might not be as easy as we first thought. Live and learn, eh?

IEEE 1003.8: Networking (IPC) Update

Steve Head <smh@hpda.hp.com> reports on the July 10-14, 1989 meeting, in San Jose, California:

Overview

P1003.8 is the IEEE POSIX committee working on network standard interface definitions for POSIX. The committee is divided into several subcommittees, including transparent file access, remote procedure call, network IPC, and MAP. This report summarizes recent activity in the network IPC subcommittee, which is currently working on two potential interfaces: a "detailed" network interface (DNI) and a "simple" network interface (SNI). DNI is roughly (though not exclusively) at the transport level. SNI is intended to be

somewhat simpler to use than DNI, but at roughly the same level.

At this meeting, a draft of DNI was begun, which included a scope, a chapter-by-chapter outline of the document specifying functionality included in each chapter, and the beginning of a rationale, which discusses goals. For SNI, goals, objects, and functionality were discussed, but without a full resolution.

Also, a schedule was adopted which forecasts the activities of the committee towards mock ballot and full ballot of DNI and SNI through January 1993.

Several joint meetings with P1003.6 (security) and P1003.4 (real time) were held on the subjects of network security and real-time IPC.

Plans were made to make P1003.8 a steering committee and to elevate each P1003.8 subcommittee (including P1003.8/2) to full POSIX committee level.

At this meeting, the main topics of discussion were:

DNI draft

A draft of DNI was begun. The draft now includes a scope, plus skeleton chapters on connection setup and tear down (including naming), data transfer, async event management, option management, POSIX 1003.1 extensions, OSI transport protocol family options, and Internet protocol family options. Appendices include related standards, a rationale, and comparisons with X/Open's XTI and BSD's sockets. Each chapter is currently language-independent, specifying functionality only, not C routines.

So far, DNI is a functional superset of XTI and sockets, although this has not been formally adopted as an explicit goal by the group.

SNI goals, objects, and functionality

The group discussed SNI goals, objects, and functionality. Some progress was made. SNI's proponents now envision it as being capable of complex operations, such as async events. Users will be able to intermix SNI and DNI routine calls as needed.

SNI may adopt some of the characteristics of UNIX standard I/O, specially tailored for networking, but the exact relationship to the UNIX standard I/O package has not yet been addressed.

Schedule

A tentative schedule was adopted for DNI and SNI.

Summer 1989 meeting

SNI goals/functionality; SNI/DNI outline

Fall 1989 meeting

SNI/DNI connection setup/teardown

Winter 1990 meeting

SNI/DNI data transfer

Spring 1990 meeting

SNI/DNI event management

Summer 1990 meeting

SNI/DNI POSIX 1003.1 extensions

Fall 1990 meeting

SNI/DNI protocol-independent options

Winter 1991 meeting

SNI/DNI miscellaneous functionality DNI protocol-dependent (ISO, ARPA, etc.) options

Spring 1991 meeting

SNI/DNI definitions

Summer 1991 meeting

SNI/DNI review drafts

Fall 1991 meeting

SNI/DNI approve drafts for mock ballot

Oct. 1991

SNI/DNI mock ballot

Winter 1992 meeting

SNI/DNI resolve mock ballot objections

Spring 1992 meeting

SNI/DNI review drafts

Summer 1992 meeting

SNI/DNI approve drafts for full use ballot

Aug. 1992

SNI/DNI full use ballot

Fall 1992 meeting

SNI/DNI resolve full ballot objections

Winter 1993 meeting

SNI/DNI resolve full ballot objections

Feb. 1993

SNI/DNI submit approved drafts to IEEE stds. board

Spring 1993

data representation network interface
goals ...

Security

We held two joint meetings with the POSIX security committee (P1003.6).

P1003.6 more or less views its role as describing necessary high-level security features and requirements, and would like to leave the job of filling in specific interfaces to P1003.8. This is agreeable to P1003.8, but both groups need to work to ensure that this division of labor leaves no holes.

Paul Melmon, of Hewlett-Packard, also made a presentation on Internet protocol address family security. The presentation covered a special interest topic, B1 security for TCP-IP networks. For this level of security, security labels are usually automatically inserted into the IP header by the system, on behalf of the process. The label content is normally determined by the security level of the process. At the receiving end, packets are rejected for reception by another process unless deemed appropriate by the system, which compares the label with the label appropriate to the receiving process. Privileged daemons such as inetd, which need to be able to handle incoming connection requests or data from processes at arbitrary levels, are an exception to this scheme. For such processes, label options need to be associated with connections and datagrams. The presentation was favorably received by the group, but no clear consensus emerged on exactly how the POSIX networking interface(s) would be impacted.

An issue emerged with respect to security and existing transport interfaces – in particular, XTI and sockets. XTI specifies Internet address family security label options based on MIL-STD 1777 version dated September 1983. 4.3 BSD allows a user to specify a choice of security label through the IPOPTIONS setsockopt() request. However, MIL-STD 1777 has been updated via RFC 1038 ("Draft Revised IP Security Option," M. St. Johns, IETF, January 1988). An even later RFC is scheduled to be released in the near future with further changes in this area. The specifications are driven primarily by needs within U.S. government agencies.

The new (RFC 1038) protocol format specification is incompatible with the old. In

addition, many vendors require a new, more extensible, IP security option for the commercial market; a consortium of vendors, including Sun, HP, Unisys, and others (at the moment, this group is called simply "the Consortium"), is addressing this need.

Also, neither XTI nor sockets specifies any restrictions on the use of label options. This may be a security hole: unrestricted users can "spoof" a higher level of security than they actually possess. For example, an "unrestricted" (low-level) process could specify that outbound data it writes to a network endpoint object be accompanied by a "classified" label, implying (to the remote system) that the data was sent by a process with a higher security level.

Finally, neither XTI nor sockets provides the ability to retrieve a label associated with an incoming UDP datagram in an atomic manner. XTI has no provisions for UDP labels at all.

In the light of these issues and recent developments DNI and SNI may need to track the standards governing security as they evolve, possibly offering a standard (and secure) interface to such features.

IPC

For historical reasons, both P1003.4 (real-time) and 1003.8 now find themselves working, independently, on IPC. We held a joint meeting with P1003.4 on IPC. The general concern was the divergent directions of the interfaces, given the overlapping user needs. There were specific differences in areas such as name resolution, options, and performance characteristics.

"Real time" IPC has two variants: one, an event-based version which simply allows passing a pointer to shared memory from one process to another; the other, a message-based version which allows data messages between sending and receiving processes. Both versions use the UNIX file system name space for rendezvousing; both versions use queues and allow various manipulations on the queues. The message-based version requires timestamps, has provisions for user-process-defined priorities and sender identification, and has several options to optimize data

transfer. In contrast, DNI and SNI are both based on a simpler, data-stream paradigm, with no queue manipulations, timestamps, filesystem rendezvousing, user-defined priorities, or sender identification, and few options for data transfer optimization. DNI and SNI may include options for message boundary delimitation, and will use a more general rendezvousing mechanism (aka name server interface) than the UNIX file system.

Unresolved issues include these:

1. Whether it is desirable to rely on a UNIX filesystem name space for general-purpose internetwork IPC rendezvous, both because machines may be far apart, and because mounting each machine's filesystems from all others is impractical in a large network.
2. How timestamp information can be kept accurate over a network.
3. How to encourage more interaction between X/Open XNET, and other concerned parties, and P1003.8. (This should require only an education process, since these groups are already interacting with P1003.4.)
4. What direction to take on the interaction of IPC and networking. The P1003.4 IPC group seems to favor generalizing the IPC mechanism for networking. This currently clashes with the networking group on transparent file access, which is currently focusing on an NFS-supportable subset of P1003.1 file semantics, and has never adopted support of P1003.4 file semantics as a formal goal.
5. Whether it is feasible, given timing and balloting considerations, to form a joint group or offload IPC onto a networking group.

It seemed generally agreed that there should be closer relations between the real-time and networking groups in this area, and that needless differences should be minimized.

One feature from real-time IPC was adopted which should allow faster performance in DNI than in either XTI or sockets: "tear-away writes." These let a user process

specify that it does not need to access a write/send buffer after a write/send operation, freeing the system to unmap the buffer from user space and schedule the buffer for DMA, thus avoiding the need for a buffer copy operation.

Naming

A name service interface working group was created at this meeting, and attracted a lot of attention, both in and out of P1003.8. We described specific needs of the DNI and SNI interfaces to the new working group at a joint meeting: simple name resolution, name registry (SNI only), and the ability to get path information for a given service. We also clarified our position that at least the simple name resolution was needed at or before the DNI/SNI full-use ballot, to avoid dependency and usability problems.

P1003.8/2 -> full POSIX committee

P1003.8/2, along with other P1003.8/x groups, is in the process of becoming a full POSIX committee (P1003.y). The P1003.8 structure will evolve to become a POSIX networking "steering committee," overseeing the efforts of each P1003.8/x group.

Steering committees are sometimes used in IEEE standards committees to structure related subgroups and join their forces whenever a concerted effort is needed to address a problem. They help ensure that redundant standards are not created and that each subgroup has a clear and unique focus. POSIX has no steering committees yet, and a minor precedent will be set if this new organization becomes formally adopted. (Other such steering committees, such as one for languages, are being contemplated and may appear in the near future.)

Language independence

The P1003 steering committee has decided that new POSIX standards (with a few exceptions) will be specified in a language-independent manner, with at least one specific language binding. (Typically, one expects this will be C.)

P1003.8 is, thus, required to comply with the P1003 steering committee decision in this

regard, and the P1003.8/x networking standards will be issued in a form that includes a language-independent specification.

Bytes versus octets

Neither POSIX nor the C standard specifies the number of bits in a byte. The number is system-dependent and accessible to a user process as CHAR_BIT, which according to the C language standard has a minimum value of 8. In networking this specification is insufficient to guarantee complete and formal interoperability, since (if an interface is specified in bytes) one system's notion of a byte may differ from another's – at least, in principle. Thus, most formal networking standards avoid the use of the term byte in favor of octet, implying an ordered set of eight bits.

POSIX data-transfer operations are defined exclusively in terms of bytes, not octets. For POSIX to be interoperable in the networking sense, either POSIX must change to octets, some relatively ugly solutions must be adopted, or some simplifying assumptions should be made whenever networking may be involved. The issue probably affects network IPC, and seems like it could also affect other areas – the most likely candidates being transparent file access and data archival.

The problem has been noted by P1003.8 at large, but not yet specifically addressed. Informal polls conducted at POSIX meetings indicate that most, and perhaps all, current vendors use eight-bit bytes. The ultimate solution may be to use weasel-wording equivalent to the assumption that interoperating systems will all use eight-bit bytes.

IEEE 1003.11: Application Transaction Processing Update

Bob Snead <bobs@ico.isc.com> reports on the July 10-14, 1989 meeting in San Jose, California:

1003.11 (application transaction processing, or TP) is one of two recently approved working groups – the other being P1003.10 (supercomputing) – whose charter is to write an application environment profile (AEP). A profile is simply a list of pointers to existing standards within the POSIX OSE (Open System

Environment). Where the group finds functionality missing from this set of standards, the group may either commission its definition by some other POSIX group or write a new PAR to request that IEEE create a standard in the area.

This was our first meeting as 1003.11; the previous three meetings were as a study group. This study group was formed last year at the Ft. Lauderdale meeting to investigate the feasibility of extending POSIX into transaction processing. In those first three meetings there was consensus that POSIX should address transaction processing.

At this point, the TP group is reviewing existing standards in detail to find out what's already been done. To this end, they have split into two subgroups, one to review models, the other to search out and review other relevant standards. There seems to be some consensus that once we understand what is available, there will still be new interfaces to define.

TP under UNIX is currently sort of a funny domain. Database vendors believe that transaction processing is theirs. They build TP primitives into their products that let application developers define transactions over modifications to data. More and more UNIX application developers want, instead, to write applications that bind a group of modifications

to data managed by assorted vendors' products, including multiple databases, screen managers, and file systems. Sensing this need, X/Open boldly chartered a group to define such services. In addition, ISO, some time ago, recognized the need for services to define transactions which span heterogeneous open systems, and began a group to define such services. ISO also has groups defining CCR (Commitment, Concurrency, and Recovery) and RDA (Remote Data Access), each of which is an essential part of TP, especially distributed TP.

Both efforts are pretty far along. X/Open has defined a model and a set of interfaces but, since they are not a real standards body, referencing their work may present some problems for P1003.11. The ISO group recently resolved all outstanding objections to their model, services, and protocols. What remains for us then is to place the relevant portions of their work into a POSIX framework, filling in the holes.

Dear Reader,

We would like to know if you found the foregoing reports informative. Should we continue to publish these in ;login:? Please send your suggestions to the editor, ellie@usenix.org.

Out-of-Print USENIX Proceedings Now Available

The Association has photocopied, bound sets of most of its past workshop and conference proceedings available for purchase.

CONFERENCE		COST	FOREIGN POSTAGE
San Diego	1983 Winter	\$28.00	\$15.00
Toronto	1983 Summer	32.00	20.00
Washington D.C.	1984 Winter	25.00	15.00
Salt Lake City	1984 Summer	29.00	20.00
Dallas	1985 Winter	15.00	10.00
Portland	1985 Summer	45.00	25.00
Denver	1986 Winter	25.00	15.00
Atlanta	1986 Summer	37.00	20.00
Phoenix	1987 Summer	35.00	20.00
Dallas	1988 Winter	26.00	15.00
San Francisco	1988 Summer	29.00	20.00
<hr/>			
WORKSHOP			
Systems Admin. I	1987 Philadelphia	4.00	5.00
Systems Admin. II	1988 Monterey	8.00	5.00
Security	1988 Portland	7.00	5.00
Graphics II	1985 Monterey	7.00	5.00
<hr/>			
NOT AVAILABLE			
<hr/>			
Delaware Conference			
Graphics I Workshop			

Orders to U.S. and Canada are shipped via printed matter. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request. Check, purchase orders, or payments by VISA/MC are accepted. (For charge orders please include card number, expiration date, and your signature.) Prepayment is required. Send orders to:

USENIX Association
 2560 Ninth Street
 Suite 215
 Berkeley, CA 94710

Reprints of individual papers from all proceedings are available for \$5.00 each; contact the Association Office.

*Proceedings Order Form***ORDER FORM****CONFERENCE & WORKSHOP PROCEEDINGS**

Qty	Proceeding		Add'l		Foreign Postage	Total
			Unit Price	Total		
—	Graphics Workshop V	Nov. '89	\$18	\$_____	\$10	\$_____
—	Distributed and Multiprocessor Workshop	Oct. '89	30	\$_____	20	\$_____
—	Large Installation Sys. Admin. III Workshop	Sept. '89	13	\$_____	9	\$_____
—	Baltimore Conference	June '89	20	\$_____	15	\$_____
—	UNIX Transaction Processing Workshop	May '89	12	\$_____	8	\$_____
—	Software Management Workshop	Apr. '89	20	\$_____	15	\$_____
—	San Diego Conference	Feb. '89	30	\$_____	20	\$_____
—	C++ Conference	Oct. '88	30	\$_____	20	\$_____
—	UNIX and Supercomputers Workshop	Sept. '88	20	\$_____	15	\$_____
—	C++ Workshop	Nov. '87	30	\$_____	20	\$_____
—	Graphics Workshop IV	Oct. '87	10	\$_____	15	\$_____
—	Washington DC Conference	Jan. '87	10	\$_____	20	\$_____
—	Graphics Workshop III	Dec. '86	10	\$_____	15	\$_____

Total price of Proceedings _____
 Calif. residents only add applicable sales tax _____
 Total foreign postage _____
 Total enclosed _____

* Discounts are available for bulk orders. Please inquire.

PAYMENT OPTIONS

Check enclosed payable to USENIX Association. Purchase order enclosed.
 Please charge my: Visa MasterCard  

Account # _____ Exp.Date _____

Signature _____

Overseas? Please make your payment in U.S. currency by one of the following:

- * Charge (Visa, MasterCard, or foreign equivalent)
- * International postal money order
- * Check - issued by a local branch of a U.S. Bank

Shipping Information

10/89

Orders to U.S. and Canada are shipped via printed matter. Please allow 2 weeks for delivery. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request.

Ship to: _____

Please mail this order form to: USENIX Association
 2560 Ninth Street
 Suite 215
 Berkeley, CA 94710

Final Printing of 4.3BSD Manuals

The 4.3BSD manuals offered by the USENIX Association[†] are now available to everyone.

The 4.3BSD manual sets are significantly different from the 4.2BSD edition. Changes include many additional documents, better quality of reproductions, as well as a new and extensive index. All manuals are printed in a photo-reduced 6"×9" format with individually colored and labeled plastic "GBC" bindings. All documents and manual pages have been freshly typeset and all manuals have "bleed tabs" and page headers and numbers to aid in the location of individual documents and manual sections.

A new Master Index has been created. It contains cross-references to all documents and manual pages contained within the other six volumes. The index was prepared with the aid of an "intelligent" automated indexing

program from Thinking Machines Corp. along with considerable human intervention from Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

While two of the manual sets contain three separate volumes, you may only order complete sets.

The costs shown below do not include applicable taxes or handling and shipping from the printer in New Jersey, which will depend on the quantity ordered and the distance shipped. Those charges will be billed by the printer (Howard Press).

To order, return a completed "4.3BSD Manual Reproduction Authorization and Order Form" to the USENIX office along with a check or purchase order for the cost of the manuals.

Manual	Cost*
User's Manual Set (3 volumes)	\$25.00/set
User's Reference Manual	
User's Supplementary Documents	
Master Index	
Programmer's Manual Set (3 volumes)	\$25.00/set
Programmer's Reference Manual	
Programmer's Supplementary Documents, Volume 1	
Programmer's Supplementary Documents, Volume 2	
System Manager's Manual (1 volume)	\$10.00

* Not including postage and handling or applicable taxes.

[†] Tom Ferrin of the University of California at San Francisco, a former member of the Board of Directors of the USENIX Association, has overseen the production of the 4.2 and 4.3BSD manuals.

4.3BSD Manual Reproduction Authorization and Order Form

This page may be duplicated for use as an order form

Purchase Order No.: _____

Date: _____

Pursuant to the copyright notice as found on the rear of the cover page of the UNIX®/32V Programmer's Manual stating that

"Holders of a UNIX®/32V software license are permitted to copy this document, or any portion of it, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included."

I hereby appoint the USENIX Association as my agent, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.3BSD Manuals as I may request.

Signed: _____

Ship to:
Name: _____

Billing address, if different:
Name: _____

Phone: _____

Phone: _____

The prices below do not include shipping and handling charges or state or local taxes. All payments must be in US dollars drawn on a US bank.

4.3BSD User's Manual Set (3 vols.)	_____	at \$25.00 each	= \$_____
4.3BSD Programmer's Manual Set (3 vols.)	_____	at \$25.00 each	= \$_____
4.3BSD System Manager's Manual (1 vol.)	_____	at \$10.00 each	= \$_____
Total	_____		\$_____

[] Purchase order enclosed; invoice required.
(Purchase orders must be enclosed with this order form.)

[] Check enclosed for the manuals: \$_____
(Our printer will send an invoice for the shipping and handling charges and applicable taxes.)

Send your check or purchase order with this order form to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Local User Groups

The Association will support local user groups by doing a mailing to assist the formation of a new group and publishing information on local groups in ;login:. At least one member of the group must be a current member of the Association. Send additions and corrections to login@usenix.org.

CA - Fresno: the Central California UNIX Users Group consists of a *uucp*-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
 Brent Auernheimer (209) 294-4373
brent@CSUFresno.edu or csufres!brent

Commercial institutions or individuals:
 Gordon Crumal (209) 875-8755
csufres!gordon (209) 298-8393

CO - Boulder: the Front Range UNIX Users Group meets monthly at different sites.

Steve Gaede (303) 447-8586
 636 Arapahoe Ave., #10
 Boulder, CO 80302

FL - Coral Springs:

S. Shaw McQuinn (305) 344-8686
 8557 W. Sample Road
 Coral Springs, FL 33065

FL - Fort Lauderdale/Miami: The South Florida UNIX Users Group meets the 2nd Tuesday of each month.

Tony Vincent, John McLaughlin (305) 776-7770
 {sun,novavax,gould}!sunvice!tony
jmcLaughlin@sun.COM

John O'Brien (305) 475-7633
gatech!uflorida!novavax!john

Don Joslyn (305) 476-6415
gatech!uflorida!novavax!rm1!don

FL - Jacksonville/Northeast: UNIX Users of Jacksonville (uujax) meets the 2nd Thursday of each month.

Tom Blakely (904) 646-2820
uflorida!unf7!tfsb

Emilie Olsen (904) 390-3621

FL - Melbourne: the Space Coast UNIX Users Group meets at 8pm on the 3rd Wednesday of each month at the Florida Institute of Technology.

Bill Davis (407) 242-4449
bill@ccd.harris.com

FL - Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.

Mike Geldner (305) 862-0949
cudas!sunfla!mike

Ben Goldfarb (305) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius (305) 869-2462
{codas,attmail}!mikel

FL - Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.

Bill Hargen (813) 530-8655
uunet!pdn!hargen

George W. Leach (813) 530-2376
uunet!pdn!reggie

GA - Atlanta: meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
 P.O. Box 12241
 Atlanta, GA 30355-2241

Marc Merlin (404) 442-4772
 Mark Landry (404) 365-8108

MI - Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.

Steve Simmons home: (313) 426-8981
scs@lokkur.dexter.mi.us office: (313) 769-4086

K. Richard McGill
rich@sendai.ann-arbor.mi.us

Bill Bulley
web@appiga.uucp

MI - Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.

Linda Mason (313) 855-4220
michigan!usr/group
 P.O. Box 189602
 Farmington Hills, MI 48018-9602

;login: 14:6

MN - Minneapolis/St. Paul: meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044

Robert A. Monio (612) 895-7007
pnessutt@nis.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105

Eric Kiebler (314) 725-9492
plus5!sluug

NE - Omaha: meets the 2nd Thursday of each month.

/usr/group nebraska
P.O. Box 44112
Omaha, NE 68144

Kent Landfield (402) 291-8300
kent@ugn.uucp

New England - Northern: meets monthly at different sites.

Peter Schmitt (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755

decvax!dartvax!nneuug-contact

NJ - Princeton: the Princeton UNIX Users Group meets monthly.

Pat Parseghian (609) 452-6261
Dept. of Computer Science
Princeton University
Princeton, NJ 08544

pep@Princeton.EDU

NY - New York City:

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

Ed Taylor (212) 513-7777
(attunix,philabs)!pencom!taylor

New Zealand:

New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

OK - Tulsa:

Pete Rourke
\$USR
7340 East 25th Place
Tulsa, OK 74129

PA - Philadelphia: the UNIX SIG of the Philadelphia Area Computer Society (PACS) meets the morning of the 3rd Saturday of each month.

G. Baun, UNIX SIG
c/o PACS
Box 312
La Salle University
Philadelphia, PA 19141
rutgers!(bpa,cbmvax)!
temvax!pacsb!(gbaun,whutchi)

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205

Jim Hummel (214) 522-2324

TX - Houston: the Houston UNIX Users Group (Hounix) meets the 3rd Tuesday of each month.

Hounix answering machine (713) 684-6590
Bob Marcum, president (713) 270-8124
Chuck Bentley, vice-president (713) 789-8928
chuckb@hounix.uucp

TX - San Antonio: the San Antonio UNIX Users (SATUU) meets the 3rd Thursday of each month.

Jeff Mason (512) 494-9336
Hewlett Packard
14100 San Pedro
San Antonio, TX 78232
gatech!petrol!hpsatb!jeff

WA - Seattle: meets monthly.

Bill Campbell (206) 232-4164
Seattle UNIX Group Membership Information
6641 East Mercer Way
Mercer Island, WA 98040
uw-beaver!tikal!camco!bill

Washington, D.C.: meets the 1st Tuesday of each month.

Washington Area UNIX Users Group
2070 Chain Bridge Road, Suite 333
Vienna, VA 22180

Samuel Samalin (703) 448-1908

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

First Class Mail

FIRST CLASS MAIL
U.S. POSTAGE PAID
Hayward, CA
Permit No. 2

Nominating Committee Report

Winter Conference Tutorials and Program

Audio I/O with the NeXT Computer

Calls for Papers

Change of Address Form

Please fill out and send the following form through the U.S. mail to the Association Office at the address above.

Name: _____ Member #: _____

OLD: _____ NEW: _____

Phone: _____

uucp: _____